GENERALIZABLE SURROGATE MODELS FOR THE IMPROVED EARLY-STAGE EXPLORATION OF STRUCTURAL DESIGN ALTERNATIVES IN BUILDING CONSTRUCTION

A Dissertation
Presented to
The Academic Faculty

by

Mehdi Nourbakhsh

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Building Construction

Georgia Institute of Technology
May 2016

GENERALIZABLE SURROGATE MODELS FOR THE IMPROVED EARLY-STAGE EXPLORATION OF STRUCTURAL DESIGN ALTERNATIVES IN BUILDING CONSTRUCTION

Approved by:

Dr. Javier Irizarry, Advisor
School of Building Construction
Georgia Institute of Technology

Dr. Lauren K. Stewart
School of Civil and Environmental Engineering
Georgia Institute of Technology

Prof. Chuck Eastman
School of Architecture
Georgia Institute of Technology

Dr. John Haymaker
Director of Research
Perkins+Will

Dr. Daniel Castro
School of Building Construction
Georgia Institute of Technology

Date Approved: January 11, 2016

To my exceptionally talented collaborator,

colleague, best friend, and wife, Samaneh

ACKNOWLEDGEMENTS

This dissertation would have not been possible without the thoughtful guidance, feedback, and support of variety of important advisors, colleagues, and friends. First and foremost, I would like to express my deepest appreciation to my advisor, Dr. Javier Irizarry, who introduced me to the joy of applied and multi-disciplinary research, for his continuous support and encouragement throughout my doctoral study.

I would like to extend my deepest gratitude to Professor Chuck Eastman for honoring me by allowing me to work with him on his research projects and making me a better researcher. I am extremely grateful to Dr. John Haymaker, who has been a creative, supportive, and invaluable mentor throughout this research. I would also like to extend my sincere thanks to Dr. Daniel Castro and Dr. Lauren Stewart for sharing their expertise and invaluable insight into this research.

I cannot begin to express my thanks to my colleagues at Autodesk. I am deeply indebted to Michael Bergin for sharing his formidable expertise and experience in computational design, optimization, and advanced manufacturing. Many thanks to Mark Davis, Dr. Adrian Butscher, Dr. Wei Lee, Hyunmin Cheong, and all of my colleagues on the OCTO Generative Design Team for their support and invaluable feedback throughout this research.

I am extremely grateful to my exceptionally talented collaborator, colleague, best friend, and wife, Samaneh, for her exuberant support, patience, and her endless love. I

would also like to thank my inspirational parents for encouraging me to follow my dreams and continue my education to the highest level.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF SYMBOLS AND ABBREVIATIONS

2/3D	2/3 Dimensional
ACI	American Concrete Institute
AISC	American Institute of Steel Construction
AM	Additive Manufacturing
ANN	Artificial Neural Network
ASCE	American Society of Civil Engineering
ASTM	American Society for Testing and Materials
CC	Contour Crafting
EA	Evolutionary Algorithm
ES	Evolution Strategy
ESO	Evolutionary Structural Optimization
FCD	Fully Constrained Design
FEA	Finite Element Analysis
Ft	Foot
GA	Genetic Algorithm
GPSA	General Purpose Structural Analysis
HOK	Hellmuth, Obata + Kassabaum
in	inch
Ksi	Kilopound per Square Inch
LMJ	Liquid Metal Jetting
MPa	Mega Pascal
NASTRAN	NASA's STRuctural Analysis

RBF	Radial Basis Function
SIMP	Solid Isotropic Material with Penalization
SIS	Inhibition Sintering

SUMMARY

The optimization of complex structures is extremely time consuming. To obtain their optimization results, researchers often wait for several hours and even days. Then, if they have to make a slight change in their input parameters, they must run their optimization problem again. This iterative process of defining a problem and finding a set of optimized solutions may take several days and sometimes several weeks. Therefore, to reduce optimization time, researchers have developed various approximation-based models that predict the results of time-consuming analysis. These simple analytical models, known as "meta- or surrogate models," are based on data available from limited analysis runs. These "models of the model" seek to approximate computation-intensive functions within a considerably shorter time than expensive simulation codes that require significant computing power.

One of the limitations of metamodels (or interchangeably surrogate models) developed for the structural approximation of trusses and space frames is lack of generalizability. Since such metamodels are exclusively designed for a specific structure, they can predict the performance of only the structures for which they are designed. For instance, if a metamodel is designed for a ten-bar truss, it cannot predict the analysis results of another ten-bar truss with different boundary conditions. In addition, they cannot be re-used if the topology of a structure changes (e.g., from a ten-bar truss to a 12-bar truss). If designers change the topology, they must generate new sample data and re-train their model. Therefore, the predictability of these exclusive models is limited.

From a combination of the analysis of data from structures with various geometries, the objective of this study is to create, test, and validate generalizable metamodels that predict the results of finite element analysis. Developing these models requires two main steps: feature generation and model creation. In the first step, involving the use of 11 features for nodes and three for members, the physical representation of four types of domes, slabs, and walls were transformed into numerical values. Then, by randomly varying the cross-sectional area, the stress value of each member was recorded. In the second step, these feature vectors were used to create, test, and verify various metamodels in an examination of four hypotheses.

The results of the hypotheses show that with generalizable metamodels, the analysis of data from various structures can be combined and used for predicting the performance of the members of structures or new structures within the same class of geometry. For instance, given the same radius for all domes, a metamodel generated from the analysis of data from a 700-, 980-, and 1,525-member dome can predict the structural performance of the members of these domes or a new dome with 250 members. In addition, the results show that generalizable metamodels are able to more closely predict the results of a finite element analysis than metamodels exclusively created for a specific structure.

A case study was selected to examine the application of generalizable metamodels for the early-stage exploration of structural design alternatives in a construction project. The results illustrates that the optimization with generalizable metamodels reduces the time and cost of the project, fostering more efficient planning and more rapid decision-

making by architects, contractors, and engineers at the early stage of construction projects.

CHAPTER 1

INTRODUCTION

This research presents a novel approach for the optimization of structures in the cloud. The first section of this chapter presents a brief history of construction and the evolution of new materials and manufacturing techniques that have changed the way of creating buildings. The chapter continues with the motivation of the research, the current state of knowledge and practice, gaps in knowledge, and research objectives and hypotheses.

A Brief History of Construction

Since prehistoric times, humans have built their houses and structures using various materials and techniques that have had an excessive impact on the evolution of the architectural and structural design of buildings. For instance, while one-story buildings with thick stone walls were once the norm, modern skyscrapers now grace skylines. These changes in material, machinery, labor, and architectural and structural design are presented in the following section.

Material

In ancient Greece, Egypt, and China, humans used stone, adobe, and timber to build their structures. The Romans used clay, stone bricks, and a type of mortar with volcanic ash known as *pozzolan*, which quickly hardened into a rigid mass. These construction materials gradually improved until the 17^{th} and 18^{th} centuries, when iron became increasingly used as beams and columns in structures. The nineteenth century witnessed the mass production of steel for structural beams and columns (Chang and Swenson 2014).

Machinery

Like material, machinery has also undergone dramatic changes over time. The ancient Greeks built jibs and cranes to lift heavy stones to upper parts of their buildings (Wright 2009). In the early 20th century, with the advent of steam-powered and internal-combustion engines, heavy machinery for construction became mass-produced. With the advent of such machinery (e.g., cranes) and elevators, people built high-rise buildings and skyscrapers with fewer laborers. As the 20th century progressed, modern manufacturing machinery used in factories enabled the casting of prefabricated structural elements such as beams and columns that were shipped to construction sites to build housing units, apartment blocks, and bridges. The use of these machines reduced the cost and completion time of construction projects, particularly heavy construction projects (Friedman 2010, Chang and Swenson 2014).

Labor

Methods of construction in the past were extremely labor-intensive. Romans, for example, had to use a multitude of slaves to build a structure. As methods of manufacturing and the quality of construction materials continually improved, the labor-intensity of construction significantly decreased. For instance, a bulldozer is ten to 100 times more productive than a construction worker. Because of these changes, contractors and owners can minimize the size of their workforce, particularly skilled workers. Thus, they have been able to lower their project costs devoted to not only material and machinery but also workers (Chang and Swenson 2014).

Impact on Architectural Design

Materials and techniques of construction also play a major role in the architectural design of buildings. For instance, after the 20th century, when the use of concrete and steel became commonplace, thick and straight walls became thin and curved, one-story

buildings became high-rises, and architects had more freedom to design the layouts of buildings. Such changes have been well-documented by Le Corbusier (1986), who stated the following:

> It was a common thing in the good old days (which still go on, alas!) to see heavy horses drawing enormous stones to the yard, and a mass of human labor unloading them, cutting and dressing them hoisting them on the scaffolding, placing them in position and, rule in hand, making lengthy adjustments to every face; such buildings might take two years to construct: to-day a building can be erected in a few months; the P.O. have recently finished their immense Cold Storage building at Tolbiac. The materials uses are confined to grains of sand and coke-breeze the size of small nuts; the walls are thin like membranes; but enormous consignments are stored in this building. Thin walls to give protection against differences of temperature, and partitions 3 to 4 inches thick in spite of the enormous loads stored there. Things have indeed altered! (p. 233).

Indeed, the construction of buildings with new materials has changed the architectural design of buildings in many ways (e.g., the elimination of some internal walls to create open spaces inside houses, the design of facades).

Impact on Structural Design

Structural engineering was not a separate discipline prior to the 1850s, when wrought-iron beams became popular in the construction of buildings. Since these building elements had to be mathematically designed and craftsmen did not have tried-and-true rules of thumb, the American Society of Civil Engineering (ASCE) rapidly

distributed technical information on records of experiments with cast and wrought iron (Friedman 2010).

In the early 1920s, the analysis of structures was a tedious task, as stated by Professor Frits Leonhardt in his book on the history of structural engineering (Leonhardt 2010):

> In the 1930's statically determinate structures were preferred, especially in case of poor foundation conditions with settlement to be expected. Single span beams were quite common. The methods to calculate the forces in statically indeterminate structures were not well-known and were tedious to be done by hand… In the computer age, numerical analysis of structures is, of course, no longer a handicap in the choice even of complicated structures. In some cases, it is not so much the calculation of static forces, but the calculation of the geometry of a construction. This was the case for the network of the Olympic Stadium in Munich (1970), for which Professor J. Argyris provided a FE program to calculate the exact length of the thousands of strands and ropes just in time (p. 6-7).

Since the mid-twentieth century, engineers have developed various structural analysis packages capable of analyzing complex structures within a short period of time. For instance, NASA funded three companies to develop a General Purpose Structural Analysis (GPSA) program known as NASA's STRuctural ANalysis (NASTRAN), capable of designing cars, ships, buildings, and even implants (Mayer 1998). In recent years, most of these structural analysis programs have been migrating from desktop computers to the cloud platform. Working in the cloud, engineers will be able to run their analysis even faster than ever before. For instance, NASTRAN can analyze 3,799,278

degrees of freedom of a car body on an eight-node cluster in about 83 minutes (HPC 2011).

Construction in the 21st Century

The fabrication of objects via the deposition of material using a print head, nozzle, or other printer technology is known as three-dimensional (3D) printing (ASTM 2012). During the past several decades, this fabrication technology has been slowly adopted by manufacturing for rapid prototyping. However, in recent years, 3D printing has been more rapidly developing for several reasons. For one, as the cost of materials used to produce 3D printers has been decreasing, the price tag of 3D printers has significantly dropped. In addition, the build quality (resolution) of 3D printers has been improving (Wohlers 2012).

Recently, 3D printing technology has been used in the construction industry to create full-scale buildings (e.g., the Canal House in the Netherlands (CanalHouse 2014), single story houses in China (Goldin 2014), and a five-story apartment building in China (Starr 2015)). Unlike buildings constructed with traditional methods of construction, 3D-printed structures can embody any shape or topology, particularly in the case of metal printing. For instance, the proposed structure of a footbridge that appears to be extremely complex will be 3D printed in the Netherlands (Figure 1).

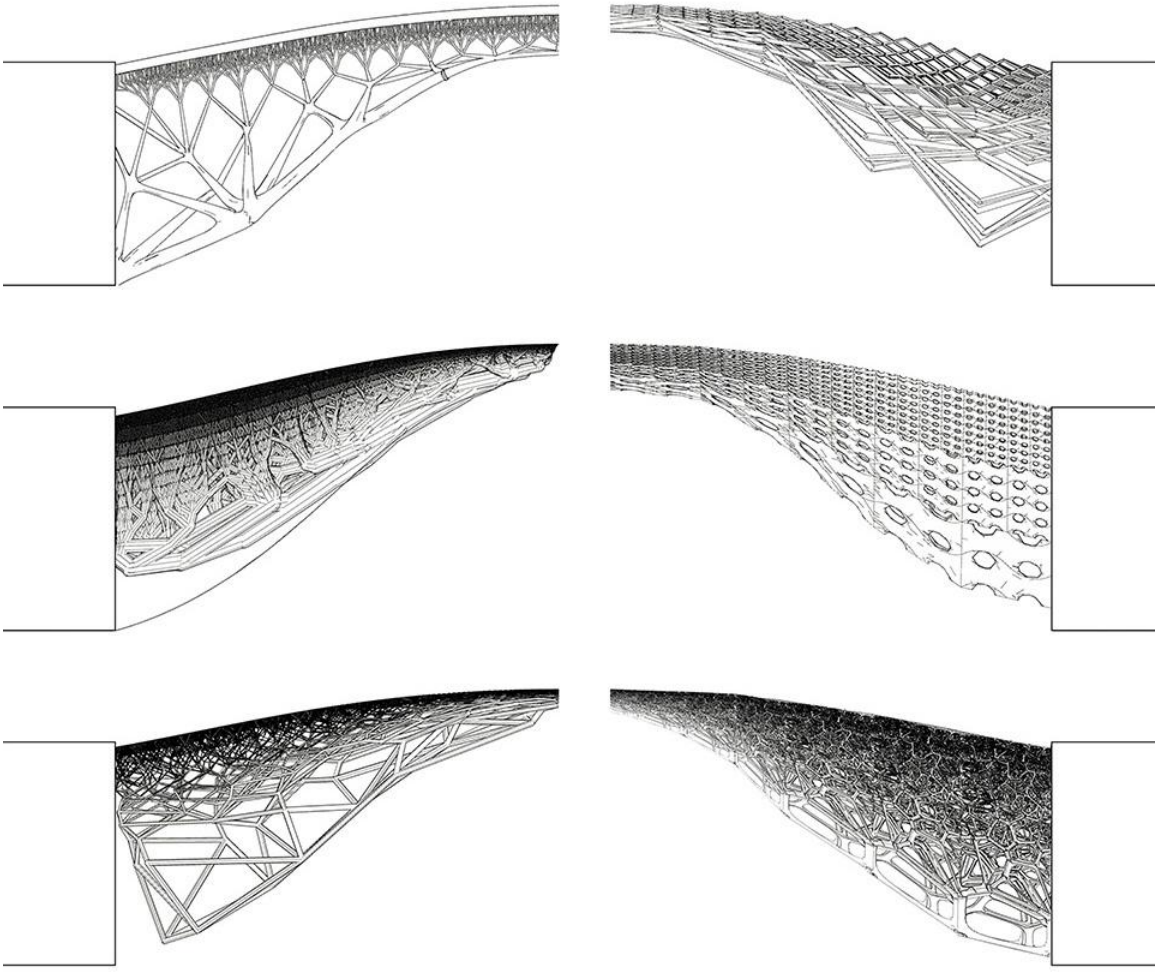


Figure 1: Proposed structure of the first 3D-printed foot bridge in the world (Mx3D 2015)
.

Motivation

The history of construction shows that whenever new materials or machinery were introduced, the architectural and structural design of buildings changed. Analogous to the 20th century, the advent of heavy machinery enabled builders to erect high-rise buildings with steel structures, in the 21st century, the advent of 3D printers are used in the architectural and structural design of buildings. For example, unlike the conventional, simple-shaped buildings typical of current building practice, future buildings may have

much more complex shapes and topologies. As 3D printers are changing the design of buildings, they are also expected to change the design of structural members. Instead of straight columns, we may design columns with a lattice structure that is both light-weight and high-performance (Figure 2). Unlike traditional columns with pre-defined shapes, the shapes and topologies of 3D-printed structures can be optimized to reduce the weight of the structure without scarifying performance. The optimization of 3D-printed structures, therefore, should substantially reduce the cost of construction.

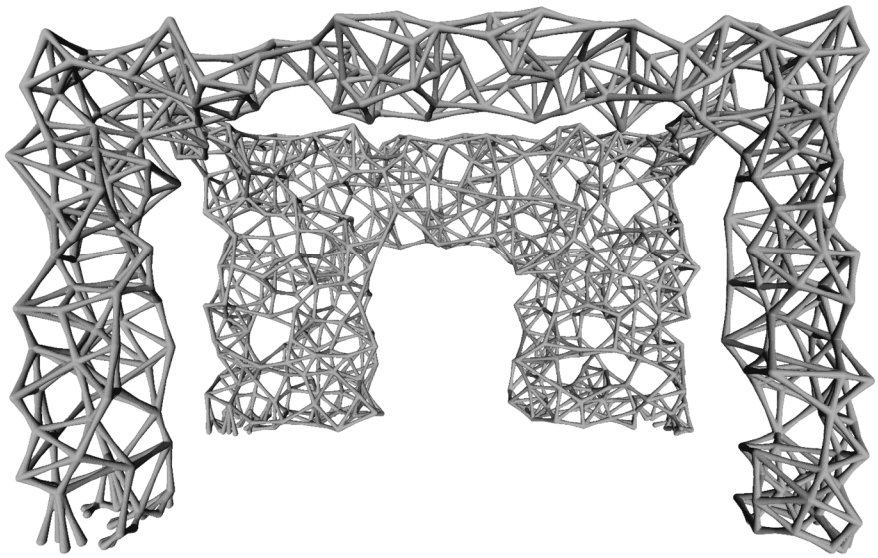


Figure 2: A conceptual structure of a future building

With new manufacturing technology, the shape of our future structures could become highly complex. The optimization of such complex structures, however, is extremely time consuming. To obtain optimization results, researchers often wait for several hours and even days after running their optimization problems. Then, if they have to make a slight change in their input parameters, they must run their optimization problem again. This iterative process of defining a problem and finding a set of optimized solutions may take several days and sometimes several weeks.

One way to reduce optimization time is to run optimization problems in the cloud. Over the past few decades, cloud and high-performance computing have significantly increased computing power with parallel processing of data for solving complex computational problems. For instance, using high-performance computing, MSC Nastran is able to analyze the hull of a ship (7,068,247 degrees of freedom) in less than 20 minutes (Nastran 2014). Even with the power of parallel computing, the optimization of complex structures, which requires thousands on runs, might be extremely time-consuming.

Another way to reduce the time of structural optimization is to predict the results of finite element analysis using approximation-based models. These simple analytical models, known as "meta- or surrogate models," are based on data available from limited analysis runs. These "models of the model" seek to approximate computation-intensive functions within a considerably shorter time than expensive simulation codes that require significant computing power. To create metamodels for structural optimization, researchers have borrowed techniques from machine learning. In one study, Salajegheh and Gholizadeh (2005) used an artificial neural network to develop a metamodel that takes in the cross-sectional area of the members of a dome and predicts the stress values of these members.

The combination of the power of cloud computing, metamodeling, and machine learning can expedite the process of the optimization of complex structures, especially structures created using additive manufacturing techniques. Running optimization problems on the cloud not only increases the speed of optimization but also facilitates the collection of massive optimization data for creating metamodels that can predict the performance of structures more reliably than those created with a limited amount of data on desktop computers.

Current State of Knowledge

One of the common techniques in building metamodels is the artificial neural network (ANN), which approximates linear and non-linear functions by mimicking biological systems. ANN models map their input onto output parameters using a set of numeric weights and activation functions. Using this technique, researchers approximated the finite element analysis of various space frames and trusses. For instance, with a ten-bar truss, Hajela and Berke (1992) demonstrated a neural network model that could predict two displacement constraints of the structure in a size optimization problem. In addition to these constraints, using the same truss, Cho et al. (2007) computed the maximum stress values of the structure. Moreover, targeting cross-sectional area of truss members, Ramasamy and Rajasekaran (1996) developed a neural network model for six types of 2D trusses: North-Light, Fink, Pratt, Howe, Quadrangle, and user-defined. The geometric properties of these trusses such as span, access type, roof slope, and spacing are the inputs of the model, which predicts the optimum cross-sectional area of truss members. In 3D structures, Kaveh and Servati (2001) used a square diagonal-on-diagonal grid to predict the optimal cross-sectional area of members based on the span length and the height of the structure. Using the same parameters in a recent study, Kamyab Moghadas et al. (2012) predicted the optimal design of a double-layer grid.

Gaps in Knowledge

All of the neural network models that appear in the literature have been designed for specific structures, so they are not generalizable to the prediction of the performance of other structures. For instance, if a model is designed for a five-bar truss, it cannot be used to predict the structural performance of another five-bar truss with a different boundary condition (e.g., applied forces). In addition, these models cannot be re-used if the topology of the structure changes (e.g., from a five-bar truss to a six-bar truss in

Figure 3). By changing the topology, new sample data should be generated and the model should be re-trained. Therefore, the predictability of these exclusive models is limited.

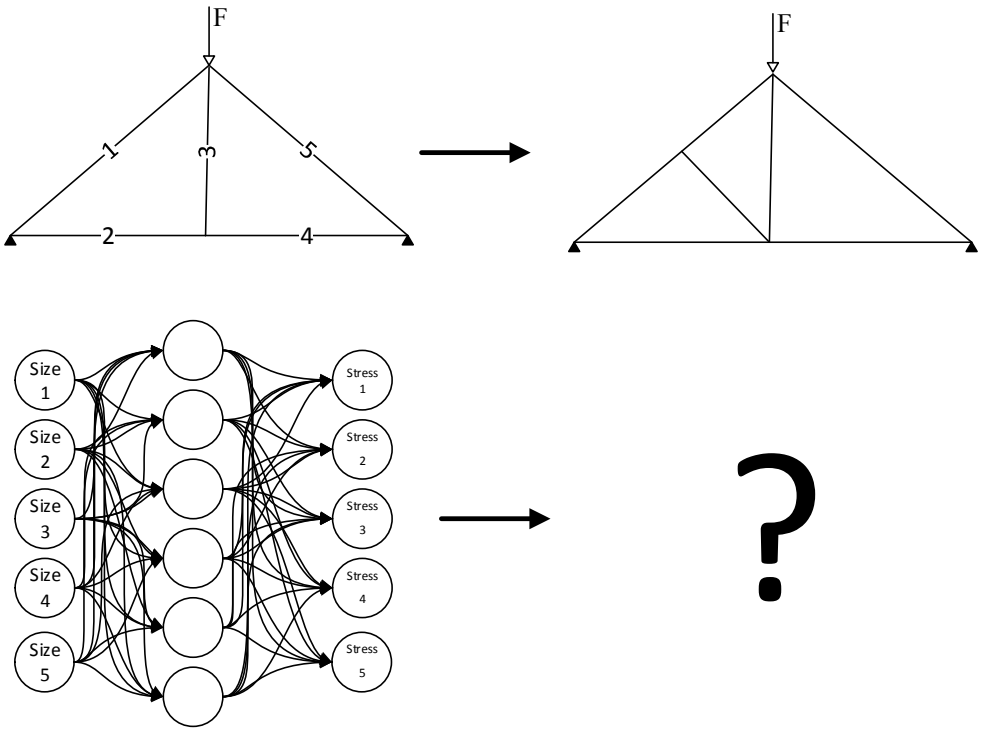


Figure 3: The generalizability of exclusive neural network models

Research Objectives

The aim of this study is to design re-usable metamodels for the early-stage exploration of structural design alternatives in building construction. To achieve this aim, the objective of this study is to perform the following tasks.

1. To create generalizable metamodels that accurately predict the performance of structures in size optimization problems
2. To test and validate the generalizability of the models using the measure of errors
3. To examine the application of generalizable metamodels for the early-stage exploration of structural design alternatives in a construction project

Research Hypotheses

Hypothesis 1

Ha: The architecture and internal weights of the neural network models of 3D-trusses and space frames differ significantly.

H0: The architecture and internal weights of the neural network models of 3D-trusses and space frames do not differ significantly.

Significance

Most of the studies that appear in the literature have examined only pinned- or fixed-joint structures. This hypothesis answers whether or not the metamodels of these structures differ. In addition, it tests the internal architecture and the mechanics of errors in these structures.

Hypothesis 2

Exclusive metamodels can more accurately predict the structural performance of space frames than generalizable metamodels.

Significance

Various researchers have demonstrated the performance of metamodels exclusively designed for a specific structure. For instance, in their studies, the input of the model was the size of each member of a given structure and the output of the model was the stress value of each member. Since a generalizable metamodel can collect data from various structures, it uses various features to encode the structures. Using these features, these models might have less accurate predictability than exclusive models.

Hypothesis 3

The results of the finite element analysis of various structures can be used for creating metamodels that predict the stress values of the members of these structures in size optimization problems.

Significance

One of the limitations of metamodels developed for the approximation of structures is lack of generalizability. Since such metamodels are exclusively designed for a specific structure, they can predict the performance of only the structures for which they are designed. In addition, in these metamodels, the data from one structure cannot be combined with the data from other structures. The combination of data from various structures is the key feature of cloud-based metamodels. This hypothesis examines whether or not these data can be combined and used for predicting the performance of structural members in size optimization problems.

Hypothesis 4

The results of the finite element analysis of various structures can be used for creating a metamodel that predicts the performance of structures within the same geometry class in size optimization problems.

Significance

One of the limitations of metamodels appearing in the literature is lack of re-usability. If a metamodel is developed for a 20-bar truss, it cannot be re-used to predict the structural performance of a 21-bar truss. In other words, if designers change the topology, they must generate new sample data and re-train their models. This hypothesis

examines whether or not metamodels can predict the performance of structures of varying topologies.

Research Scope

To address these hypotheses, we create and test metamodels using four types of structures in three classes: a horizontal slab, a vertical wall, and a symmetric dome with predefined loads. The magnitude and direction of loads are fixed to avoid the "curse of dimensionality." We analyze these steel-frame structures with continuous circular cross sections in linear static mode with isotropic material. Therefore, the analysis of structures for anisotropicity is not within the scope of this research.

Organization of the dissertation

To increase the readability of the dissertation, we present the methodology and results of each objective in a separate chapter (Figure 4). Therefore, Chapter 3 presents how to create generalizable metamodels (objective 1), Chapter 4 shows the methodology and results of the four hypotheses (objective 2), and Chapter 5 demonstrates the application of generalizable metamodels in a construction project (objective 3).

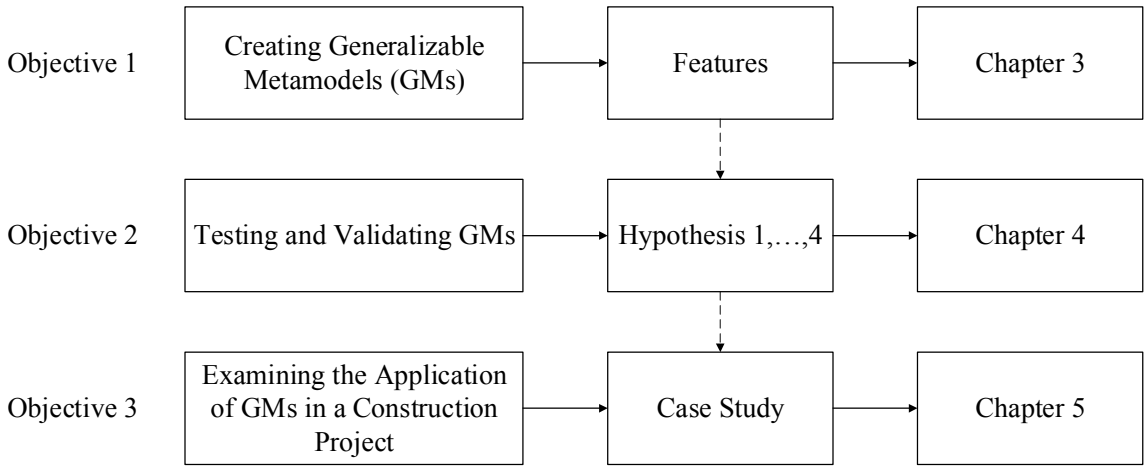


Figure 4: Dissertation organization

CHAPTER 2

LITERATURE REVIEW

This chapter briefly reviews the most relevant literature for the structural optimization of 3D-printed buildings. It elaborates on three main topics: the generation of geometries and the analysis and the optimization of structures.

3D Printing of the Analysis of Structures

The architectural and construction industries are applying large-scale additive manufacturing (AM) in three main ways: contour crafting (CC) (Khoshnevis 1999), concrete printing (Lim, Le et al. 2009), and D-shape (Dini 2014). Contour crafting uses extrusion and troweling of cement-based materials to create smooth, ruled surfaces (Zhenghao and Behrokh 2009) (Figure 5). D-shape uses a powder deposition process in which each layer is laid and bound sequentially (Dini 2014) (Figure 6). Concrete printing uses the same material and extrusion technique, but at a higher resolution than CC, and it also can print in 3D space (Lim, Buswell et al. 2012) (Figure 7).

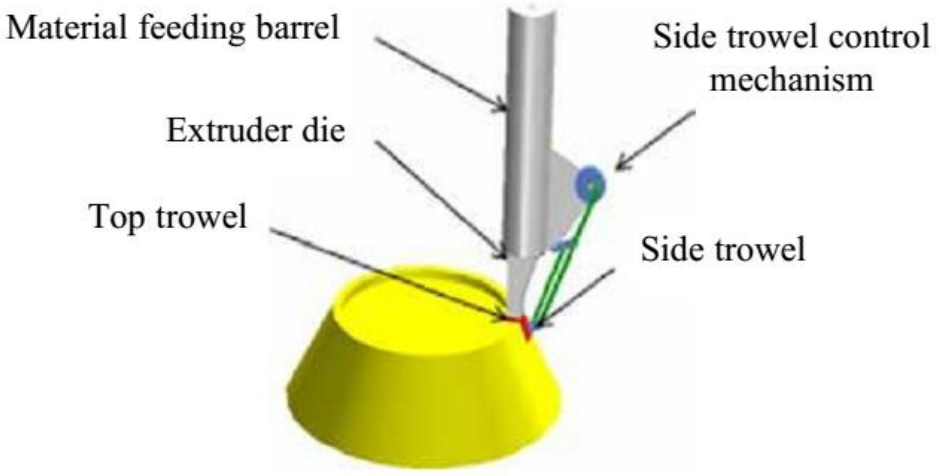


Figure 5: Principle of contour crafting (Zhenghao and Behrokh 2009)

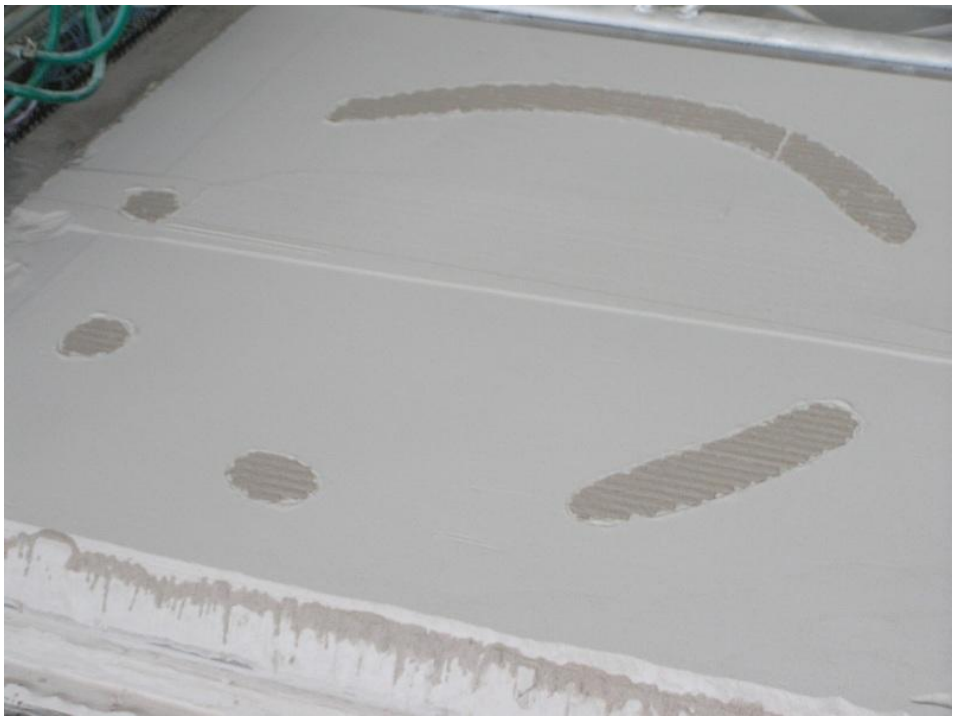


Figure 6: First printed layer of Radiolaria (Dini 2014)

Figure 7: Concrete printing (Lim, Le et al. 2009)

In addition to these three techniques, a Dutch company is using metal sintering to print a 3D footbridge on a canal in the Netherlands using a six-axis robotic arm (Mx3D 2015) (Figure 8). Unlike the other stationary 3D printing techniques, the robots can move on the bridge to print and extend the structure. These welding robots continuously add drops of welds onto the draw rods of steel in any shape (Figure 8 and Figure 9).

Figure 8: Possible design of the foot bridge (Mx3D 2015)

Figure 9: A 3D-printed object using robotic arms (FastCodeSign 2015)

The materials being used in AM construction processes are granular (Dini 2014), concrete (Lim, Le et al. 2009), mortar (Khoshnevis 2004), plastic (CanalHouse 2014), or metal (Mx3D 2015). Some of these materials use binders, which make them steady. For example, the D-shape process involves adding a chlorine-based liquid granular materials. Except for its use in the recent effort devoted to printing the footbridge, metal printing has not been used in the construction of building parts because laser - and electron beam-based printers are prohibitively expensive. For this reason, researchers have attempted to introduce technologies that lower the cost of high-performance metal printers. For instance, researchers at the University of Texas, Arlington, developed a liquid metal jetting (LMJ) process in which individual molten droplets are ejected and then connected (Priest, Smith et al. 1997). In addition, a recent study at the University of Southern California explored the application of selective inhibition sintering (SIS) to a low-cost metal additive manufacturing machine (Torabi, Petros et al. 2014).

Printing buildings with these materials has initiated a new era of the construction industry. Walls, floors, structures, façades, doors, and windows can be built within a short period of time in customized dimensions with bespoke materials. In addition, the amount of material in these building components could decrease by optimizing the most costly elements in buildings (e.g., the structure). Unlike the traditional method of construction, in which pre-defined sections are used, 3D-printed structures potentially consist of members with continuous sizes. Therefore, the optimization of these structures becomes a continuous problem. In addition, the cost of the material used in 3D-printed structures has a more direct relationship with the weight of the structure than that used in the traditional method of construction because of the way that 3D-printers use materials to build a structure.

Structural Optimization

Maximizing utility using limited resources is known as optimization (Kirsch 1993). Structural optimization problems are grouped in three main categories: topology optimization, shape optimization, and size optimization. The description of each category is as follows.

Topology Optimization

The goal of topology optimization is to find the optimal distribution of material that satisfies a series of constraints (e.g., loading conditions, support, boundary conditions) within a design domain (Bendsoe and Sigmund 2003). Topology optimization, which involves identifying the shapes, number, and location of holes in a structure and the connectivity of the domain (Bendsoe and Sigmund 2003), is divided into two types: discrete and continuous. In discrete structures, the objective is to find the optimum number of members, their positions, and their connectivity (Bendsøe 1995). In continuous structures, the shape of external and internal boundaries and the number of inner holes are simultaneously optimized (Eschenauer and Olhoff 2001).

A popular technique in topology optimization is solid isotropic material with penalization (SIMP) (Bendsøe 1989). In this method, the design domain is discretized and the material property of each section is controlled (Bendsøe and Sigmund 1999). One limitation of this method is numerical instability such as mesh dependency (Sigmund and Petersson 1998). Another popular group of methods is evolutionary structural optimization (ESO) (Xie and Steven 1993). These methods iteratively remove unnecessary materials while evolving towards the optimum or simultaneously add materials to the most demanding areas and remove superfluous ones from other areas (bi-directional) (Querin, Steven et al. 1998).

Shape Optimization

The goal of shape optimization is to determine the shape of the boundaries of structural components. In contrast to other optimization strategies, shape optimization employs geometric parameters as optimization variables (Bletzinger, Firl et al. 2010). One popular technique in shape optimization is the concept of the design element, in which structures are divided into small blocks, and subsets of these blocks are altered towards the optimal solution (Imam 1982).

Size Optimization

The goal of sizing optimization is to find the thickness distribution of structural members in order to minimize or maximize a physical quantity (e.g., peak stress, deflection) and to satisfy all constraints (Bendsoe and Sigmund 2003). The size of a structure can be optimized using single- or multi-level methods. Single-level methods involve a single stochastic or deterministic optimization algorithm. Multi-level algorithms employ more than one algorithm, each operating on a set of variables (Flager, Adya et al. 2014). One bi-level study using multi-level algorithms instead of conventional algorithms on a roof truss of a 30,000 seat arena (590 ft.) shows estimated cost savings of $1.2 million USD (Flager, Adya et al. 2014).

Structural Optimization Steps

The optimization of structures consists of various steps, starting from defining an optimization problem to determining the best set of design solutions (Gane and Haymaker 2012). This section discusses only two steps of structural optimization: geometric representation and optimization algorithms.

Geometric Representation

One of the first steps of optimization is to select a geometric representation. The type of geometry depends on the category of optimization (topology, shape, size) and the

availability of numerical solvers. One of the most common geometric representations is triangular mesh (Ding 1986). Meshes without sharp angles (less than 10 degrees) are compatible with most finite element analysis packages. Shapes can also be defined using their boundaries. Boundary-based methods define boundaries either explicitly (e.g., using polynomials, splines) (Ding 1986) or implicitly (e.g., iso-contours) (Allaire, Jouve et al. 2002, Allaire, Jouve et al. 2004). One of the disadvantages of these methods is the inefficiency of computing optimal solutions. To reduce the complexity of structural boundaries, scientists introduce a scalar function referred to as level set. Maintaining the level set function in the optimization process using a distance function is computationally much cheaper than maintaining mesh representations (Sigmund and Petersson 1998).

Optimization Algorithms

Deterministic Algorithms

Deterministic algorithms commonly require an initial configuration and a gradient of a constraint function. The transition between one state to another is determined by the gradient function, which can be calculated using the first derivative of the constraint and the objective of the problem (Flager, Soremekun et al. 2014). Examples of deterministic algorithms are linear programming (Topping 1983), sequential linear programming (Pedersen and Nielsen 2003), and the Lagrange multiplier method (Imai and Schmit Jr 1981).

A specialized deterministic algorithm that operates on discrete sizing variables is the fully constrained design method (FCD) (Flager, Soremekun et al. 2014). Unlike deterministic methods, FCD does not require the calculation of the first derivative of the constraint and objective function, demonstrating its flexibility. In addition, the search function of the FCD is independent of the objective and constraint function, demonstrating its generalizability. The algorithm is also not limited to the continuity of

search space, showing its scalability. The method was demonstrated to be scalable to structures including more than 100 sizing variables within a time frame comparable to that of conventional design practice (Flager, Soremekun et al. 2014).

To solve practical cost minimization problems, scientists have used a hybrid deterministic approach (Kripakaran, Gupta et al. 2007). Most heuristic algorithms use a random initial states and attempt to minimize randomness as they iterate. To minimize randomness in the initial stage, scientists developed a novel approach that uses clustering techniques to generate a reliable initial solution. This step is followed by a local search for the optimal solution. The proposed approach was able to generate higher quality (lower cost) geometries much faster than that of genetic algorithms.

Stochastic Algorithms

Stochastic algorithms do not require the gradient of objective functions, but instead they use probabilistic transition rules. These methods are effective at finding solutions even though they are computationally more costly than deterministic algorithms (Hasançebi, Çarbaş et al. 2009).

Simulated Annealing

This method mimics the annealing process of physical systems by establishing an analogy between the cost of an objective function and the energy level of physical systems (Kirkpatrick, Gelatt et al. 1983). In this Monte Carlo simulation process, the initial energy of the system changes, and if a change produces negative energy (e.g., cooling down), the new state will be accepted. The technique is used to calculate the tradeoff between the number of member groups and the minimum mass of the structure (Shea, Cagan et al. 1997). This method has successfully generated trusses with optimally directed topologies, numbers of groups, and sizes of each group. In a 3D layout space, simulated annealing was used to find the optimum layout of 3D components (Cagan,

Degentesh et al. 1998). The results demonstrate the capability of simulated annealing to generate alternatives and quality solutions within a reasonable amount of time.

Evolutionary Algorithms

Techniques that simulate nature in a crowd of individuals evolving towards optimum generations are referred to as evolutionary algorithms (EAs) (Hasançebi, Çarbaş et al. 2009). Core streams of EAs, genetic algorithms (GAs), have also been commonly used in structural optimization because they are effective at solving complicated real-world problems. For instance, an early study of the shape optimization of a truss structure developed a fuzzy-GA algorithm that significantly reduced computational time and enhanced search efficiency (Soh and Yang 1996).

A recent study optimized the shape, the configuration, and the size of a truss using evolution strategies (ESs) with continuous and discrete variables. The optimal topology configuration was achieved by employing a design domain approach and ES (Hasançebi 2007).

Particle Swarm

The principle behind this algorithm is based on the movement of animals (e.g., birds, fish), in which the behavior of individuals is affected by either the best local or global individual (Coello, Van Veldhuizen et al. 2002). Scientists extended this algorithm to handle multiple objectives. An example is optimizing only one objective at a time (i.e., a dynamic neighborhood) (Xiaohui and Eberhart 2002) or storing the non-dominated individuals that include unconstrained elite archives (Fieldsend and Singh 2002).

Structural Optimization of Trusses and Space Frames

This section presents a review of a well-studied area of research, structural optimization. It presents several studies pertaining to the optimization of trusses and space frames and highlights some of the limitations of these studies.

The objective of optimization in most structures is to reduce either the weight or the cost of structures, the latter of which is considered a function of the weight. Since the cost of structures in practice depends on several criteria such as constructability, the repetition of design, and the procurement of material (Gustafson 2010), studies using objective functions have not taken the complexity of structures into account. Therefore, some structures with an excessive number of section types are more expensive than some slightly heavier structures with fewer section types (Liu, Burns et al. 2006).

To reduce the complexity of structures, most researchers have decreased the number of design variables by categorizing them into a few groups. If, for instance, a 1,955-bar truss and a 181-bar truss have 34 and 17 groups of section sizes, respectively (Flager, Adya et al. 2014, Flager, Soremekun et al. 2014), the grouping of members significantly reduces not only the complexity of structures for construction but also the time of optimization. For example, using stochastic optimization techniques, the optimization of a 1,955-bar truss with 1,000 sections per variable is extremely time consuming compared to a 1,955-bar truss with 50 sections per variable. These optimization techniques and the reduction in the number of variables are acceptable because of the limited number of section sizes, a constructability constraint. In a future that enables the 3D printing of structures, we will not have these constraints.

Given the thousands of nodes and members in structures with continuous section sizes, the process and techniques of structural optimization are expected to change. For instance, instead of using desktop computers, researchers or engineers will be more prone to optimizing their structures using the power of cloud computing. In addition, optimization techniques that are cloud compatible (e.g., scalable) will be favored approaches. Running optimization problems on the cloud has several advantages. Besides the obvious advantages of cloud computing such as time and cost savings, cloud computing provides a central repository of a vast number of optimization problems and solutions. This cloud of data can be used to create a metamodel that can predict the

results of optimization problems. The following section presents a current review of metamodeling techniques for structural optimization problems.

Table 1: Summary of the literature review on the structural optimization of trusses and space frames

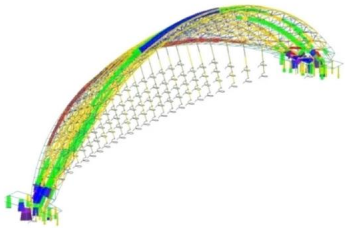
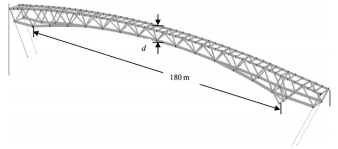
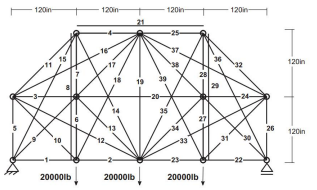
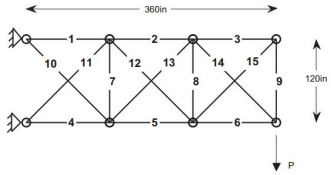
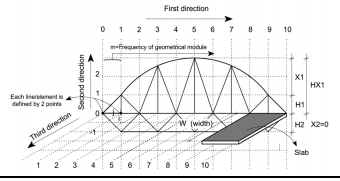
No	Citation	Structure Type	Use	No. of Variables	Range of Variables	Continuous or Discrete	Using Metamodel	Structural Optimization Type	Structural Optimization Algorithm	Optimization Objectives	Screenshot of Structures
1	(Flager, Soremekun et al. 2014)	3D Truss	Roof	1,955-bar truss (34 groups)	10-30 sections per variable	D	No	Size	Fully Con-strained Design	Weight	
2	(Flager, Adya et al. 2014)	3D Truss	Roof	181-bar truss (17 groups)	10-30 sections per variable	D	No	Shape+ Size	Bi-Opt	Weight	
3	(Miguel, Lopez et al. 2013)	3D Truss		39-bar truss	-0.225 - 2.25 in^2	C	No	Topology Shape+ Size	Firefly	Weight	
4	(Miguel, Lopez et al. 2013)	3D Truss		15-bar truss	32 per variable	D	No	Shape+ Size	Firefly	Weight	
5	(Khodadadi and von Buelow 2014)	3D Truss	Bridge	85-bar truss			No	Topology	Non-Destruc-tive Dynamic Popula-tion GA	Weight	

Table 1 (continued)

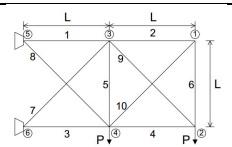
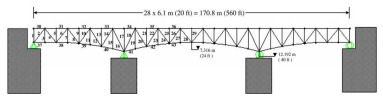
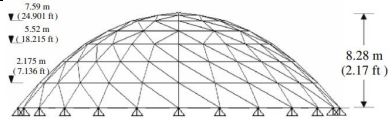
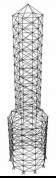
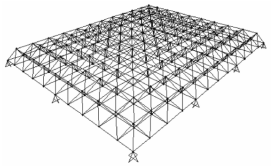
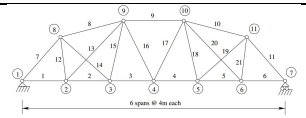
No	Citation	Structure Type	Use	No. of Variables	Range of Variables	Continuous or Discrete	Using Metamodel	Structural Optimization Type	Structural Optimization Algorithm	Optimization Objectives	Screenshot of Structures
6	(Mueller 2014)	2D Truss	Bridge	Varies <25			No	Topology	GA	Cost	
7	(Allison, Khetan et al. 2013)	2D Truss		10-bar truss			No	Topology+ Shape	GA	Weight	
8	(Hasançebi, Çarbaş et al. 2009)	Plane Truss	Bridge	113- bar truss	43 design variables	D	No	Size	Various stochastic algorithms	Weight	
		Braced truss	Dome	354-member truss	22 variables	D	No	Size	Various stochastic algorithms	Weight	
		Space truss	Tower	582-member truss	32	D	No	Size	Various stochastic algorithms	Weight	
		Double-layer grid	Grid	960-member grid	251	D	No	Size	Various stochastic algorithms	Weight	
9	(Kripakaran, Gupta et al. 2007)	2D Truss	Bridge	Varies, 21-bar truss	210	D	No	Size	Clustering and local search	Cost	

Table 1 (continued)

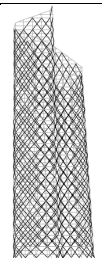
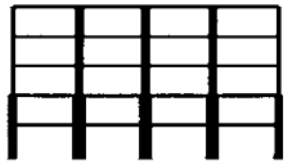
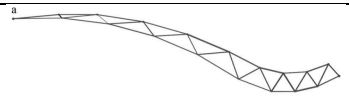
No	Citation	Structure Type	Use	No. of Variables	Range of Variables	Continuous or Discrete	Using Metamodel	Structural Optimization Type	Structural Optimization Algorithm	Optimization Objectives	Screenshot of Structures
10	(Baldock, Shea et al.)	Bracing	Building	250m-high building			No	Topology	Search	Number of bracing elements	
11	(Liu, Burns et al. 2006)	Space Frame	Building	Five-story four-bay	5 variables for beams 15 for columns	D	No	Size	GA	Weight + Com-plexity	
12	(Shea, Aish et al. 2005)	3D Truss	Roof	73m-wide span		D	No	Topology+ Shape +Size	Simulated annealing	Weight	
13	(Kicinger, Arciszewski et al. 2005)	Bracing	Building	5 bays 30 stories			No	Topology+ Shape +Size	Various evolution-ary algorithms	Weight	
		Bracing	Building	3 bays 26,32,36 stories			No	Topology	GA	Weight	

Table 1 (continued)

No	Citation	Structure Type	Use	No. of Variables	Range of Variables	Continuous or Discrete	Using Metamodel	Structural Optimization Type	Structural Optimization Algorithm	Optimization Objectives	Screenshot of Structures
14	(Shea and Cagan 1999)	2D Truss	Roof	Varies			No	Topology+ Shape + Size	Shape annealing	Weight	

Metamodeling

Estimating the performance of design objectives instead of computing them is known as surrogate modeling (Queipo, Haftka et al. 2005), or metamodeling. This technique approximates design objectives by determining the continuous function of design variables from a limited set of data. The optimization process using metamodeling entails various steps (Forrester and Keane 2009). After the selection of optimization variables, several initial samples, the results of costly analysis operations, should be analyzed. Then, a metamodel should be selected and fitted into the modeling data. The model should be validated and assessed for accuracy before it is used. Afterwards, a search function should find new designs for analysis.

Use of Metamodels

Metamodeling can be used in engineering optimization for model approximation, design space exploration, and problem formulation, and as support for various optimization types (Wang and Shan 2006). The following section briefly describes each use case.

Model Approximation

The most common use of metamodels is for predicting the results of costly analysis that requires significant computing power. Based on data available from limited analysis runs, such models seek to predict results within a considerably shorter period of time than simulation code (Forrester, Sobester et al. 2008).

Design Space Exploration

At the early stage of design, designers must more thoroughly understand the relationship between their design inputs and objectives by exploring the design space. For instance, by constructing a design space using inexpensive metamodels, they can investigate the effect of design variables and constraints on the design objective.

Problem Formulation

After gaining a deeper understanding of the relationship among variables, constraints, and objectives, designers may change their input or output parameters, that is, they reformulate the problem. Metamodels can help with formulating an optimization problem that is more accurate and easier to solve.

Optimization Support

Metamodels are intended to solve a variety of costly optimization problems such as multi-objective optimization, multi-disciplinary design optimization, and probabilistic design optimization.

Metamodeling Techniques

Sampling Plan

The first step in the design of metamodels is to develop a sampling plan. During this step, designers should plan to select a limited number of designs of a design space for analysis with simulation codes. Although an increased number of samples would improve the accuracy of the metamodels, it would also increase the computational time required to analyze the model. Therefore, selecting an efficient sampling technique is crucial to the success of predictive models.

In classic methods originating from the theory of Design of Experiments, designers pre-select their sampling points so that the evaluation of their hypothesis becomes independent of random errors in their physical experiments (Wang and Shan 2006). Among these methods, one of the most convenient sampling technique is "full factorial," in which designers split the design space into rectangular grids from where they uniformly pick their points. To improve this approach, designers generate random sub-samples within each grid to ensure a uniform projection of samples on each axis. This method is called "stratified random sampling," the basis for Latin square and random Latin hypercube. In the Latin square, a square made of n*n design variables are created and filled with (1, 2, 3,…, n) so that each number appears only once in each row or column. The Latin hypercube is a multi-dimensional extension of the Latin square (Forrester, Sobester et al. 2008).

Metamodeling Choice

Even though studies have presented various metamodeling techniques such as the polynomial (linear, quadratic, or higher), the spline (linear, cubic, NURBS), kriging, radial basis functions (RBF), the decision tree, the random forest, and the support vector machine, they have reached no consensus about which model is superior to the others (Wang and Shan 2006). This study uses the artificial neural network, a metamodeling technique that will be explained in the following section.

Artificial Neural Network

The neural network was first introduced in biological systems for mathematically representing information processing (McCulloch and Pitts 1943). Later on, the technique was broadly used among scientists and scholars for pattern recognition. This section

describes a specific class of artificial neural networks, multilayer perceptron, which has proven to have considerable practical value (Bishop 2006).

A multilayer perceptron (MLP) is a feed-forward neural network model is a nonlinear function with a vector of input parameters (x_i), output parameters (y_k), adjustable control parameter w, and non-linear basis function φ_j (x):

$$y_k(x, w) = f\left(\sum_{j=1}^{M} w_j \varphi_j(x)\right). \quad \text{Equation 1}$$

The goal of constructing a neural network model is to replace the basis function with a set of parameters that can be adjusted during the training process. In other words, using an M linear combination of input parameters, a series of functional transformations known as activations (a_j) are constructed and transformed using a differentiable, nonlinear activation functions such as a sigmoid.

$$a_j = \sum_{j=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}$$

$$w_{ji}^{(1)} = weights$$

$$w_{j0}^{(1)} = biases \quad \text{Equation 2}$$

(1) *indicates weights and biases in the first layer of the network*

$j = 1, \ldots, M$ *and* M *represent the total number of inputs*

Similarly, for the last layer of a two-layer network, we obtain

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} z_j + w_{k0}^{(2)}$$

$$w_{ji}^{(2)} = weights \quad \text{Equation 3}$$

$$w_{j0}^{(2)} = biases$$

(2) : *weights and biases in the second layer of the network*

$k = 1, \ldots, K$ *and* K *represent the total number of outputs*

$z_j = h(a_j)$ z_j *is the activation function of the first layer*

Using an appropriate activation function for the final layer, we can compute the outputs of the model:

$$y_k = \sigma(a_k), \qquad \text{Equation 4}$$

where

$$\sigma(a) = \frac{1}{1+\exp(-a)}. \qquad \text{Equation 5}$$

The following equation represents a neural network model with a sigmoidal activation function (Bishop 2006).

$$y_k(x, w) = \sigma\left(\sum_{j=1}^{M} w_{kj}^{(2)} h\left(\sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_{k0}^{(2)}\right). \qquad \text{Equation 6}$$

One way to compute the adjustable parameters (w) is to use a set of input (x_n) and target (t_n) vectors as a training set. The objective is to fit a curve that satisfies the input and output parameters, minimizing the following error function in regression problems.

$$E(w) = \frac{1}{2}\sum_{n=1}^{N} ||y(x_n, w) - t_n||^2 . \qquad \text{Equation 7}$$

Minimizing the error function is an iterative process with the adjustment of the weight matrix in a sequence of steps from the front to back layers. This method is called "back propagation," one of the most efficient methods of computing the derivative of the error function with respect to the weight (Bishop 2006).

Metamodeling-based Design Optimization Strategies

Metamodels are used in the optimization problems using three techniques. Traditionally, engineers design global metamodels and use them as surrogates of expensive objective functions. This sequential approach requires a large data set of examples, but it may not allow systematic model validation. Another technique is an adaptive strategy that allow both validation and optimization in a loop: Designers update and validate the metamodel as they evaluate more samples in the optimization loop. The last approach, direct sampling, uses metamodels as a guide for adaptive sampling and excludes them from the optimization loop (Wang and Shan 2006).

The curse of dimensionality

If the number of design variables increases, the time it takes to find samples of metamodels increases—the so-called "curse of dimensionality." If a variable is sampled in each bin of a one-dimensional design space with 'n' equally divided bins, the variable should be sampled in n^k times in k-dimensional space to achieve the same sampling density (Forrester, Sobester et al. 2008).

Reviews of Metamodels Developed for Structural Analysis or Optimization

Neural network models of various 2D trusses have been introduced and examined in several studies (Table 2). With a ten-bar truss, Hajela and Berke (1992) demonstrated a neural network model that predicts two displacement constraints of a ten-bar truss in a size optimization problem. In addition to these constraints, using the same truss, Cho et al. (2007) computed the maximum stress values of the structure. Targeting the cross-sectional area of truss members, Ramasamy and Rajasekaran (1996) developed a neural network model for six types of 2D trusses: North-Light, Fink, Pratt, Howe, quadrangle,

and user-defined. The geometric properties of these trusses, such as span, access type, roof slope, and spacing, are the inputs of the model, which predicts the optimum cross-sectional area of truss members.

Several researchers have examined more complex structures such as 3D grids and space frames. For example, Kaveh and Servati (2001) used a square diagonal-on-diagonal grid to predict the cross-sectional area of members based on the span length and the height of the structure. Using the same parameters in a recent study, Kamyab Moghadas et al. (2012) predicted the optimal design of a double-layer grid.

Since most of the research in the literature examined either simple or complex structures, the members of which are grouped into few categories, their neural network models have only one or two hidden layers with few neurons. For instance, one study grouped the structural members of a 1,300-bar grid space dome into 26 categories (Salajegheh and Gholizadeh 2005). Similarly, a study of a 288-bar grid denoted three categories for all structural members.

Table 2: Summary of studies on metamodeling

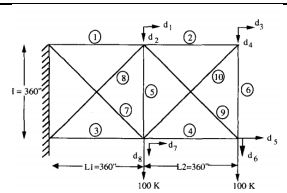
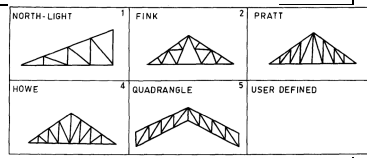
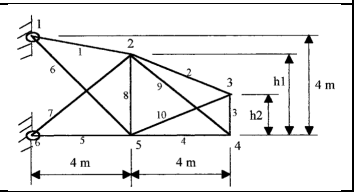
Reference	Application	Structure	Architecture	Input	Output	Training	Screenshot
(Hajela and Berke 1992)	Size optimization	10-bar truss	10-6-6-2	Cross-sectional area	Displacements of two nodes	100	
(Ramasamy and Rajasekaran 1996)	Size optimization	Five types of Roof trusses 15- to 33-bar trusses	4-4-4 4-4-4 4-4-4 4-4-6 6-6-6	(1) Span of the truss (2) Access type (3) Slope of roof (4) Spacing of truss Truss 5 has two additional inputs. (5) Angle of bottom chord; (6) Distance between the parallel chords at the end.	Cross Sectional Area	66 63 47 47 41	
(Jenkins 1999)	Weight	10-bar plane truss	#joints * DOF of each joint	Nodal loads (normalized)	Displacement	-	

Table 2 (continued)

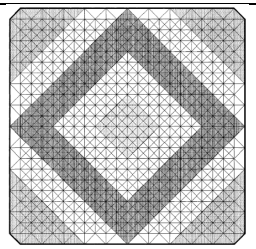
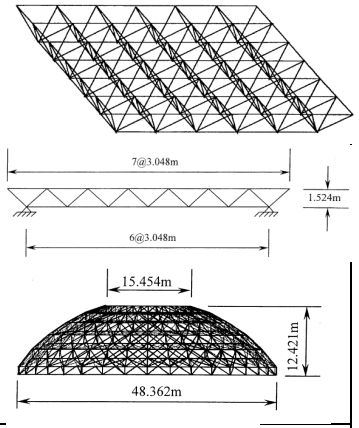
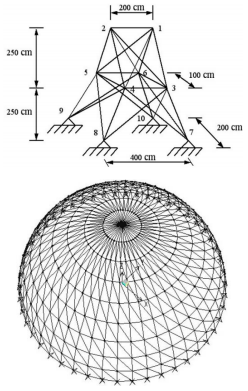
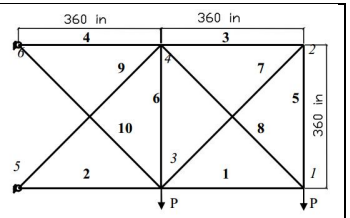
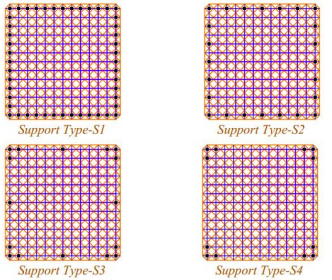
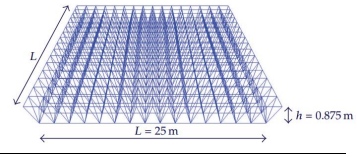
Reference	Application	Structure	Architecture	Input	Output	Training	Screenshot
(Kaveh and Servati 2001)	-	Double-layer grid	2-X1-X2-1 1<X1<6 1<X2<7	Span length, Span height	Weight	219	
(Kaveh and Servati 2001)	-	Square diagonal-on-diagonal grids		Span length, span height, region no.	Cross-sectional area	219	
(Tashakori and Adeli 2002)	Weight	3 space trusses: 1) 85N/288M/3G 2) 64N/231M/3G 3) 432N/1548M/44G,	CPN Input/ competition/ interpolation /output later	Discrete cross-sectional properties	Effective cross-sectional properties	Varies	
(Salajegheh and Gholizadeh 2005)	Weight	1) 25-bar space tower 2) 1,300-bar grid space dome (xN/1300M/26G)	CPN	Cross-sectional area	stress		

Table 2 (continued)

(Cho, Xia et al. 2007)	Weight optimization	10-bar truss	16-20-3	Cross-sectional area	Maximum stress and deformation at two nodes	90	
(Vasanwala , Desai et al. 2008)	Weight	Double-layer Grid	6-X1-4-1 (RBF) X1: 30,35,40, 60,65 6-X1-X2-1 (BP) X1: 4 or 6 X2: 2,3,4	Length (L), depth of grid (H), and support conditions (S)	Minimum grid weight	80 * 4	
(Kamyab Moghadas, Choong et al. 2012)	Weight	Double-Layer Grid	2-?-18	Length and depth of the grid	Deflection/ optimal design		

Summary of the Literature Review

All of the neural network models appearing in the literature have been designed for specific structures, so they are not generalizable to the prediction of the performance of other structures. For instance, if a model is designed for a ten-bar truss, it cannot be used to predict the structural performance of another ten-bar truss with a different boundary condition. In addition, these models cannot be re-used if the topology of the structure changes (e.g., from a ten-bar truss to a 12-bar truss). Changing the topology requires that new sample data be generated and the model be re-trained. Therefore, the use of these exclusive models presents severe limitations.

Cloud computing has reshaped information technology processes because of the scalability of computing and the availability of data. All data in the cloud are available at any time, and users can use current data, re-use achieved data, or track the history of data. In addition to the availability of data, the time it takes to process data decreases significantly because of parallel processors are used on various clusters.

The combination of cloud computing and artificial intelligence has launched a new era in computing. Companies such as Google, Amazon, and Facebook have compiled massive databases, instituted high-performance computing, and developed unique artificial intelligence algorithms that can spontaneously predict the behavior of their users based on a history of data. As more and more people interact with their services, these prediction models become smarter. In other words, these systems store past data and re-use them to create predictive models that become faster and more reliable over time.

Despite all these available technologies, no study has examined re-using past data on the cloud instead of direct analysis for predicting the performance of structures. To have a cloud of data, we should be able to combine the results of analysis from various structures and then predict the performance of structures based on their boundary

conditions. This study answers the question of whether or not the structural analysis of data from various structures can be combined and reused for creating metamodels that can infer the performance of structures.

CHAPTER 3

CREATION OF GENERALIZABLE METAMODELS

Two main phases of the methodology of this study are feature generation and model creation and verification (Figure 10). The goal of the feature generation phase is to create a set of feature vectors that encode various geometries of space frames or trusses into machine-readable codes. Using samples of feature vectors generated in this phase, the aim of the model creation phase is to develop, test, and verify neural network models that predict structural performance (the results of a finite element analysis) of various structures. These phases are described in the following sections.

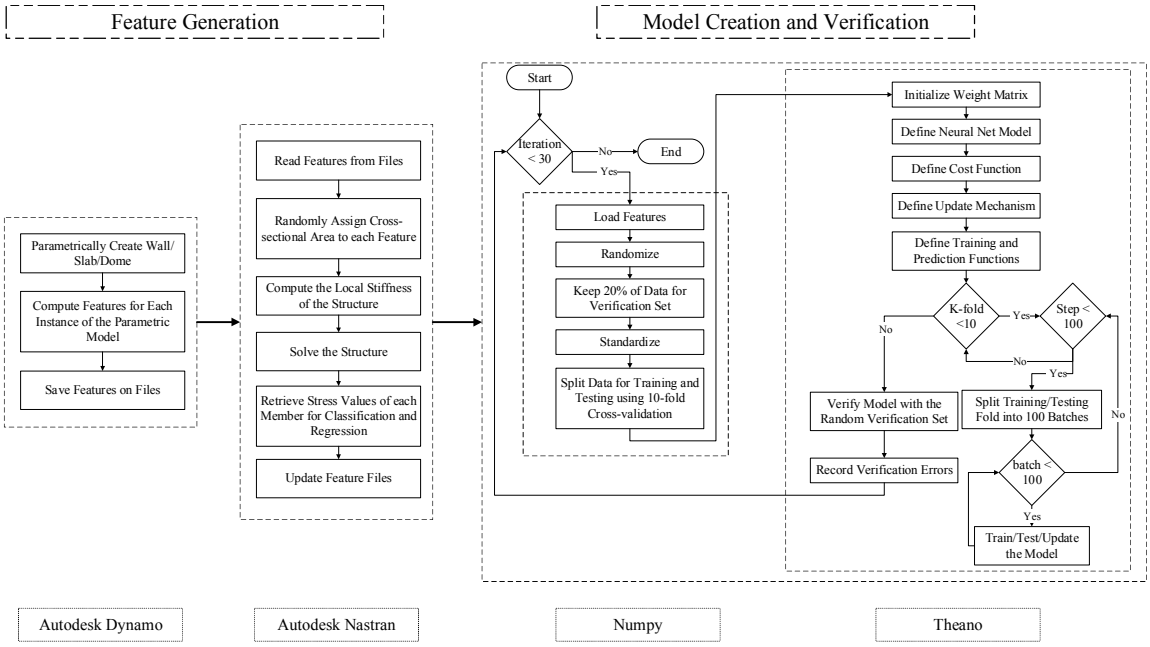


Figure 10: Research methodology

Phase 1: Feature Generation

Neural network models can approximate any function if various samples of the input and output of the function are available. These samples, or numerical values, have a

property or the properties of structures known as “features.” To generate these features, a parametric model of a geometry was created in Autodesk Dynamo. Second, some features of the geometry were extracted from the instance of the parametric models. The geometry was then imported to Autodesk Nastran, and the structure was solved using finite element analysis (FEA). Finally, the results of this analysis were recorded as the label of the features. Figure 11 illustrates the feature generation process, and the following section describes the following two steps of this phase.

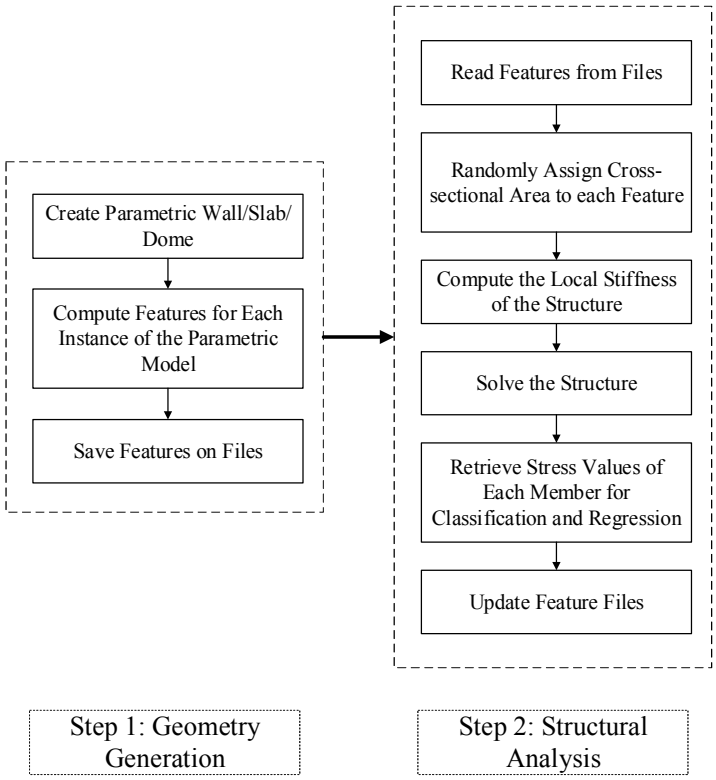


Figure 11: Feature generation

Step 1: Geometry Generation

Definition of "Geometry Class"

The goal of this step is to parametrically generate geometries for structural analysis and generate samples for neural networks. To demonstrate the generalizability of the model, we chose three classes of structures: a horizontal slab, a vertical wall, and a spherical dome. By changing one variable of the parametric models of each geometry class, we generated four instances of each class.

Create Parametric Wall/slab/dome

1) Dome: A geodesic dome with the radius of 18 m (59.05 ft.) was designed using Autodesk Dynamo. Maintaining the same radius but changing the density of members, we generated 250-, 700-, 980-, and 1,525-member domes. Figure 12 shows the geometry of the 700-member dome, and Figure 13 illustrates a portion of the Dynamo graph for generating the dome. In these figures, the spheres represent the location of the support in the dome.

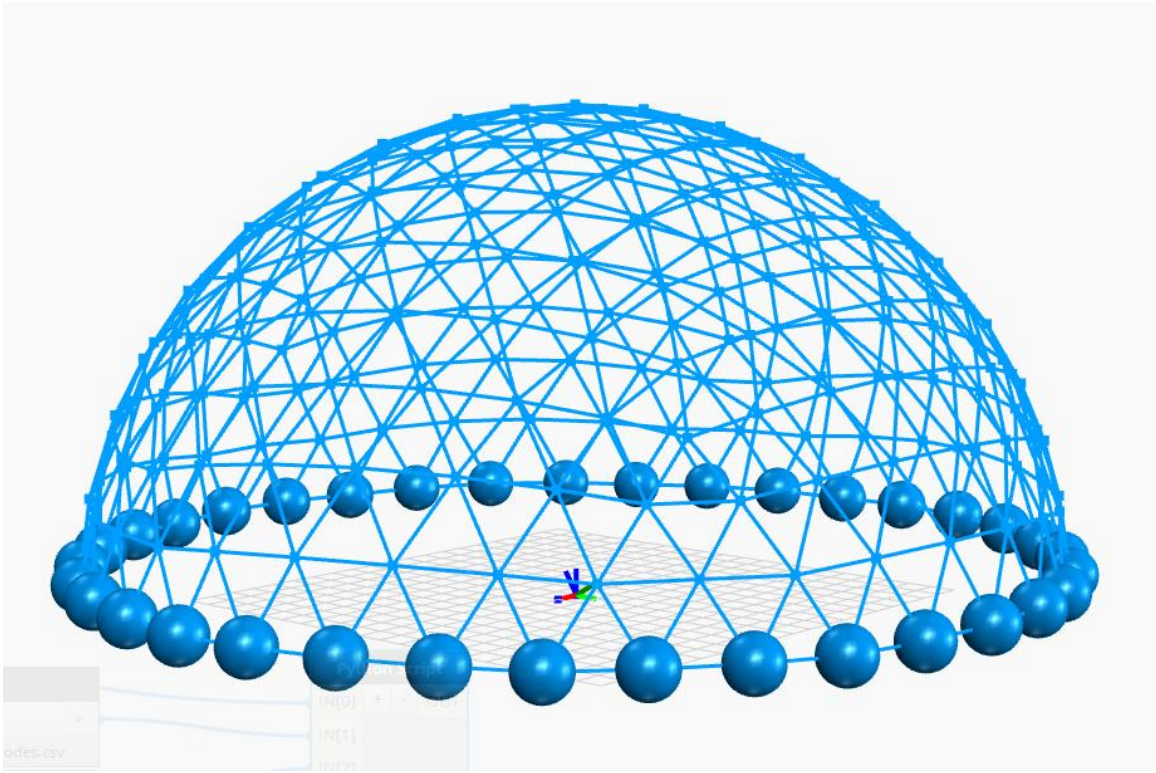


Figure 12: Geometry of the 700-member dome

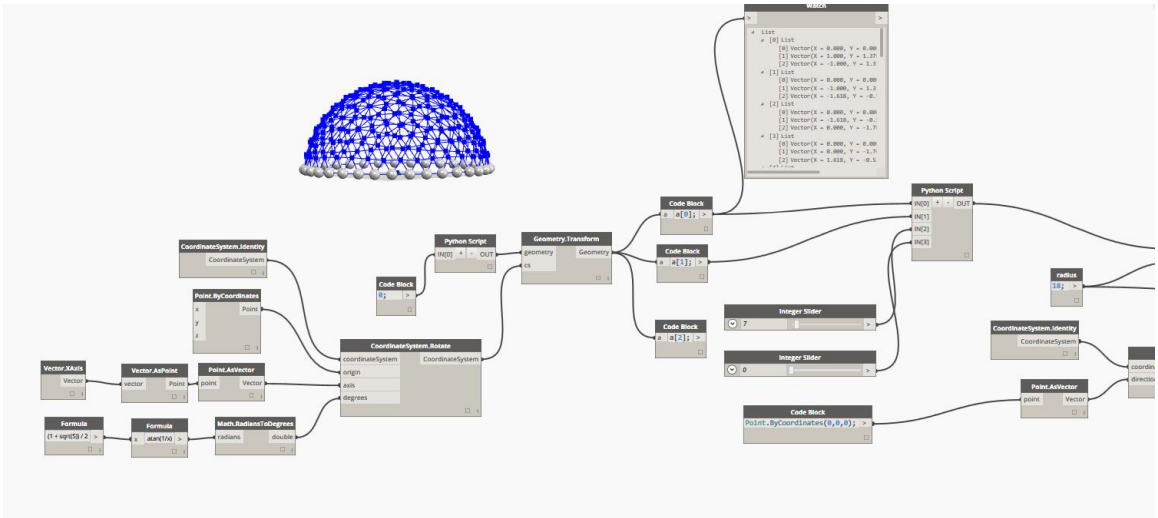
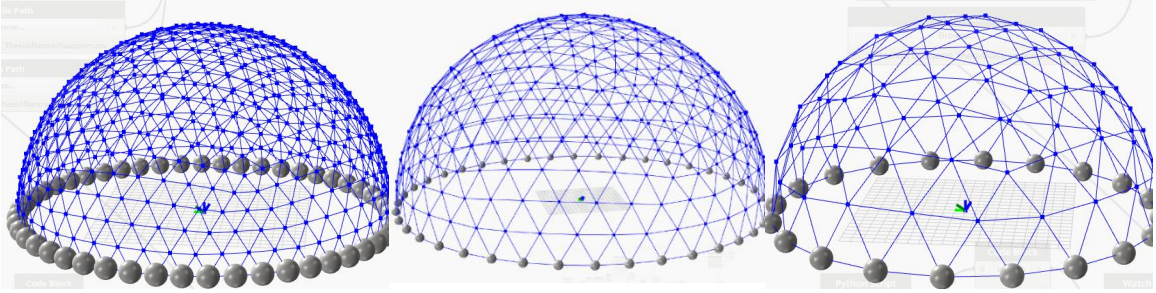


Figure 13: Dynamo graph for generating the 700-member dome



1525-member 980-member 250-member

Figure 14: Various domes generated for sampling

2) Slab: The term “slab” in this study refers to a two-level fixed- or pinned-joint grid. The unit of the grid is a 10m*10m*10m (32.80 ft.* 32.80 ft.* 32.80 ft.) cube (Figure 15). By increasing the number of these cubes in the X and Y directions, we generated 361-member (six cube*six cube), 784-member (nine cube*nine cube), 961-member (ten cube *ten cube), and 1,600-member (13 cube *13 cube) slabs using Autodesk Dynamo (Figure 17).

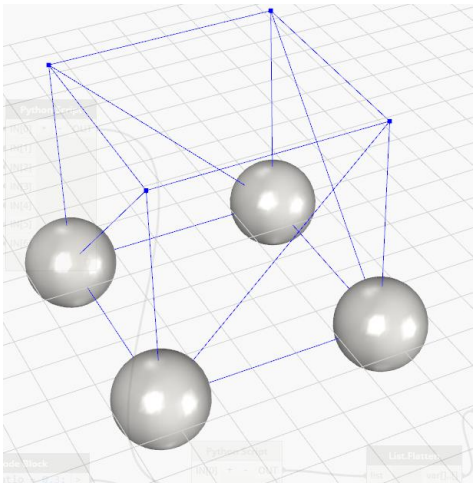


Figure 15: The unit of the slab

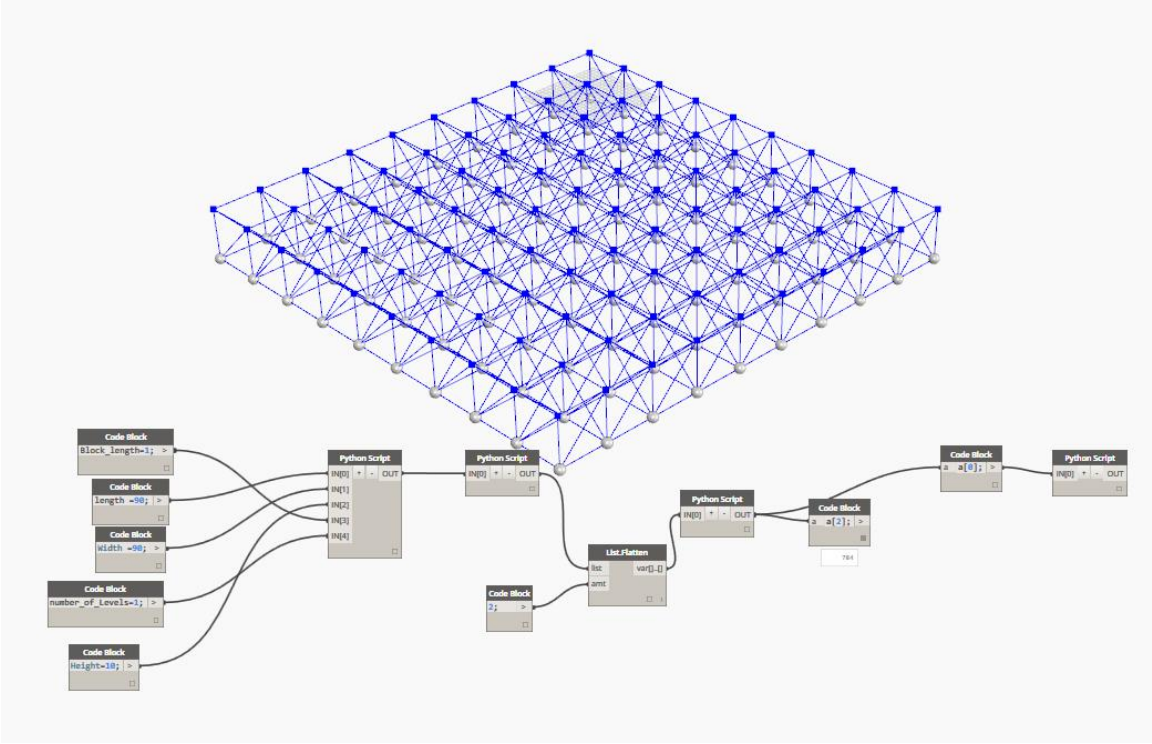
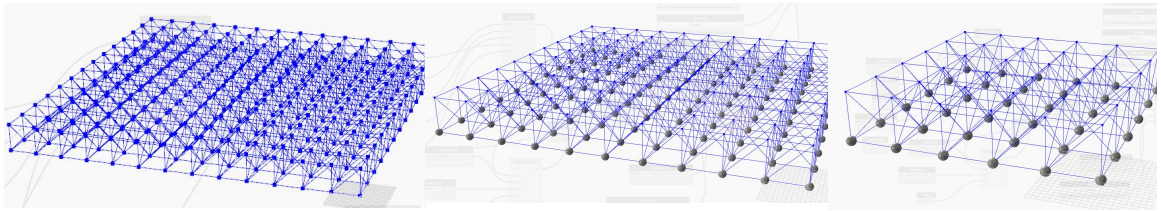


Figure 16: Dynamo graph for generating the 784-member slab



1600-member 961-member 361-member

Figure 17: Various slabs generated for sampling

3) Wall: The term "wall" in this study refers to a number of 5m*5m*5m grids that are stacked on top of one another. The input parameters of the wall are the number of blocks in the X and Z directions, the length, the width, and the height of the structure (Figure 18). However, to create an instance model of the wall in this study, we kept five cubes for the height and varied the number of cubes in the X direction. Using Dynamo, we created 304-member (six cubes * five cubes), 736-member (15 cubes* five

cubes), 928-member (19 cubes * five cubes), and 1,504-member (31 cubes * five cubes) walls (Figure 19).

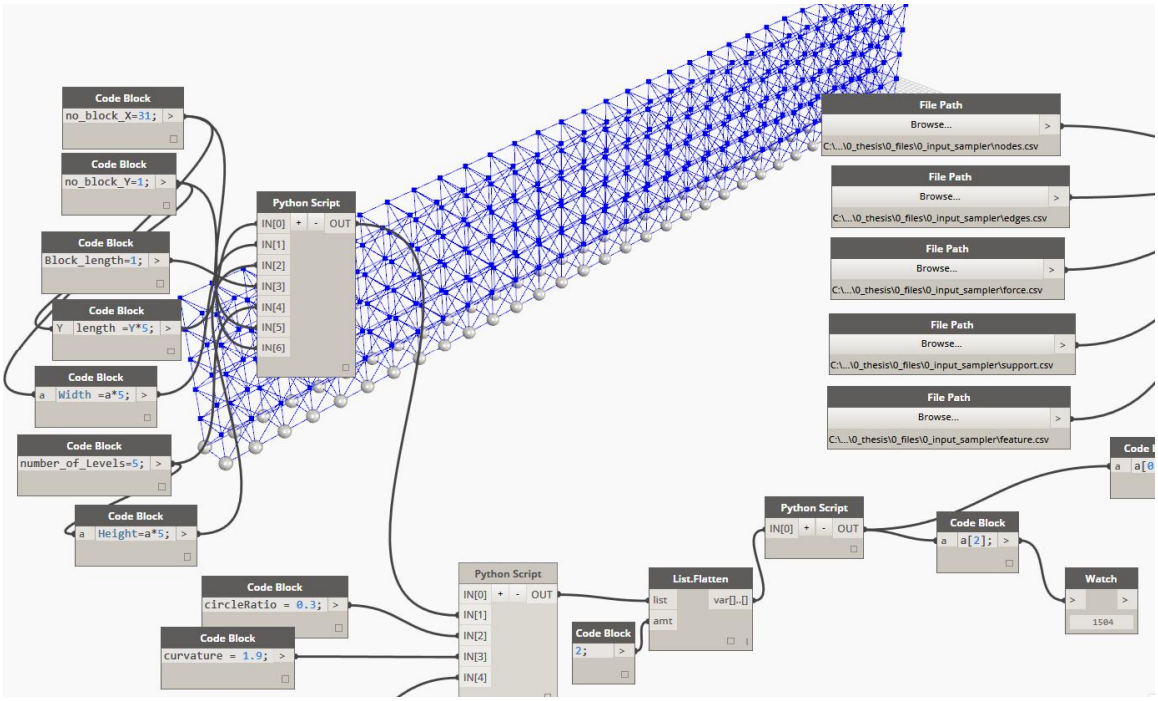
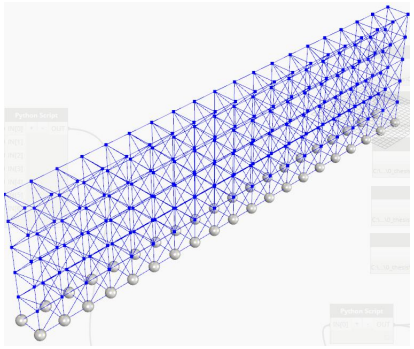
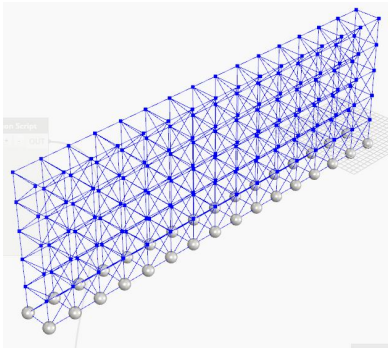


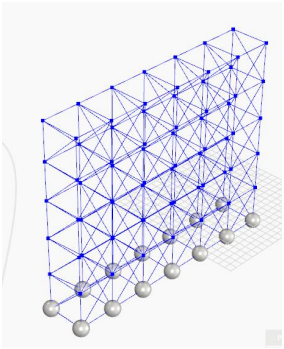
Figure 18: Dynamo graph for generating a 1,504-member structural frame



928-member



736-member



304-member

Figure 19: Various structural frames generated for sampling

Compute Features for Each Instance of the Geometry

In order to have a global descriptor of all geometries, we introduce ten unique features in this study. Some of these features are related to the nodes, and others are related to the members. Features describing the nodes are used twice in each feature vector since each member of the structures has two connected nodes –a left node and a right node– and features describing members (e.g., cross-sectional area) are used only once in each vector. The following table shows a list of node features calculated for various geometries in Dynamo.

Table 3: List of all feature descriptors for nodes

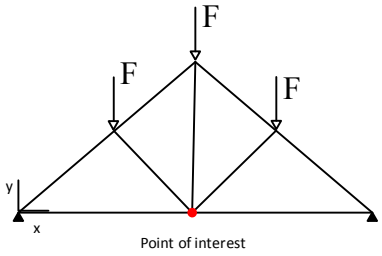
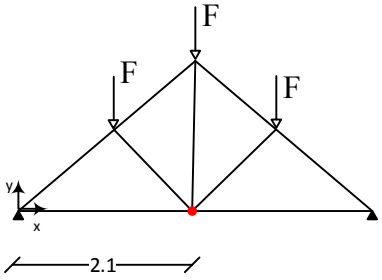
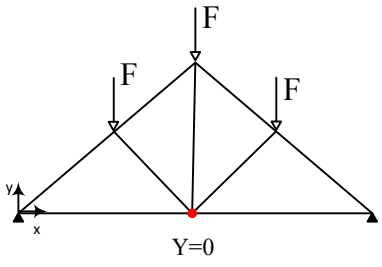
	Features	Description	Graphical Representation	Value
1	Joint type	Type of the connection of node		1
2	X node	x coordinate system of node		2.1
3	Y node	y coordinate system of node		0.0
4	Z node	z coordinate system of node		NA

Table 3 (continued)

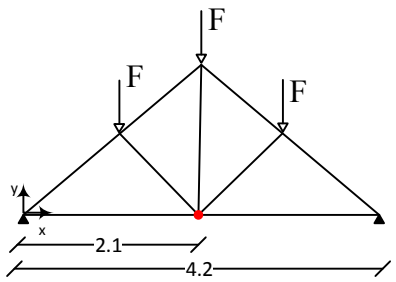
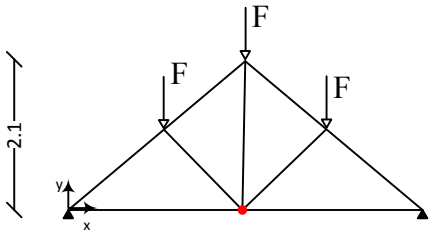
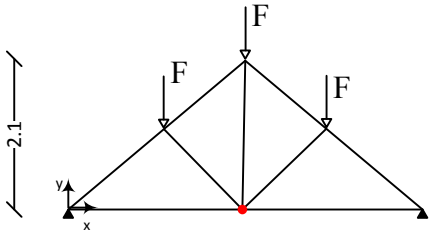
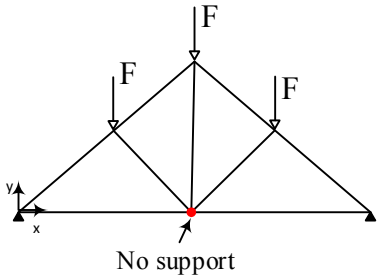
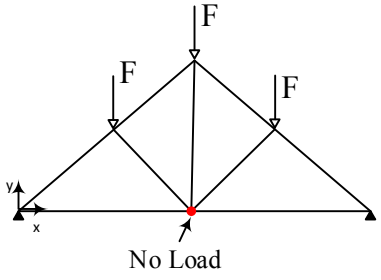
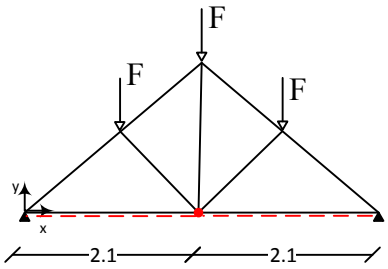
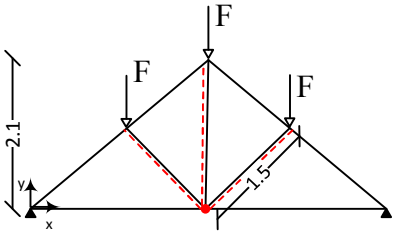
5	X / X max	Relative placement of node in the structure in X direction		0.50
6	Y / Y max	Relative placement of node in the structure in Y direction		0.0
7	Z / Z max	Relative placement of node in the structure in Z direction		NA
8	Support on the node (0/1)	Whether or not there is a support on the node		0
9	Load on the node (0/1)	Whether or not there is a load on the node		0

Table 3 (continued)

10	Proximity to supports	Inverse of the distance to all supports		1/4.2
11	Proximity to loads	Inverse of the sum of the distance to all forces		1/5.1

The first feature, joint type, describes the type of joint of a node. A list of joint types is presented in Table 4. The joint types of the first three rows of the table are from the dome class, and the joint types of the last three rows of the table are from the wall and slab structure. The distribution of these joint types differs from structure to structure. For example, 78% of joints are type 3, 16% type 1, and 6% type 2 in a dome with 250 members (Figure 20).

Table 4: Graphical representation of joint types

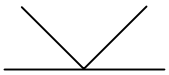
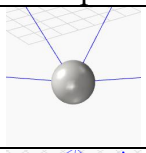
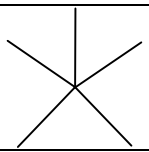
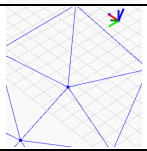
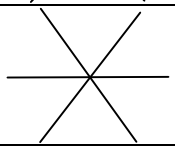
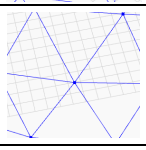
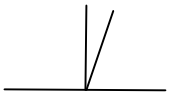
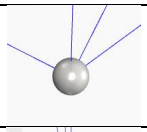
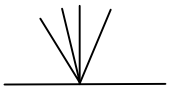
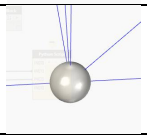
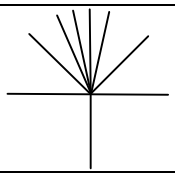
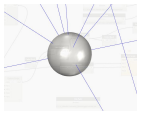
Joint Type	2D Representation	Graphical Representation
1		
2		
3		

Table 4 (continued)

4		
5		
6		

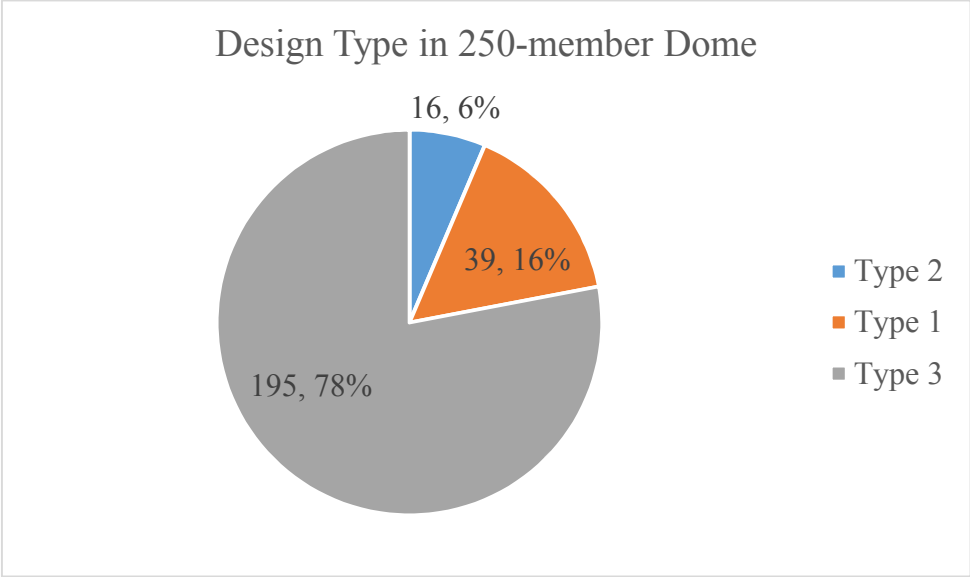


Figure 20: Distribution of joint types in a 250-member dome

Each member of these structures has two connected nodes: a left and a right node. Each of these nodes is described using 11 descriptors, presented in Table 3. Figure 21 shows feature vectors describing two nodes connected to each member of the 361-member slab. A visualization of these features is presented in Figure 22 to 25 for the dome, the slab, and the wall.

Joint Type	X node	Y node	Z node	X / X max	Y / Y max	Z / Z max	Support?	Load?	Support dist	Load dist	Joint Type	X node	Y node	Z node	X / X max	Y / Y max	Z / Z max	Support?	Load?	Support dist	Load dist
3	0	0	0	0.00	0.00	0.00	1	0	11.10	1.09	4	10	0	0	0.20	0.00	0.00	1	0	11.29	1.23
4	10	0	0	0.20	0.00	0.00	1	0	11.29	1.23	5	10	10	0	0.20	0.20	0.00	1	0	11.52	1.40
5	10	10	0	0.20	0.20	0.00	1	0	11.52	1.40	4	0	10	0	0.00	0.20	0.00	1	0	11.29	1.23
4	0	10	0	0.00	0.20	0.00	1	0	11.29	1.23	3	0	0	0	0.00	0.00	0.00	1	0	11.10	1.09
3	0	0	10	0.00	0.00	0.91	0	1	1.09	11.10	4	10	0	10	0.20	0.00	0.91	0	1	1.23	11.29
4	10	0	10	0.20	0.00	0.91	0	1	1.23	11.29	5	10	10	10	0.20	0.20	0.91	0	1	1.40	11.52
5	10	10	10	0.20	0.20	0.91	0	1	1.40	11.52	4	0	10	10	0.00	0.20	0.91	0	1	1.23	11.29
4	0	10	10	0.00	0.20	0.91	0	1	1.23	11.29	3	0	0	10	0.00	0.00	0.91	0	1	1.09	11.10
3	0	0	0	0.00	0.00	0.00	1	0	11.10	1.09	3	0	0	10	0.00	0.00	0.91	0	1	1.09	11.10
4	10	0	0	0.20	0.00	0.00	1	0	11.29	1.23	4	10	0	10	0.20	0.00	0.91	0	1	1.23	11.29
5	10	10	0	0.20	0.20	0.00	1	0	11.52	1.40	5	10	10	10	0.20	0.20	0.91	0	1	1.40	11.52
4	0	10	0	0.00	0.20	0.00	1	0	11.29	1.23	4	0	10	10	0.00	0.20	0.91	0	1	1.23	11.29
3	0	0	0	0.00	0.00	0.00	1	0	11.10	1.09	4	10	0	10	0.20	0.00	0.91	0	1	1.23	11.29
4	10	0	0	0.20	0.00	0.00	1	0	11.29	1.23	5	10	10	10	0.20	0.20	0.91	0	1	1.40	11.52
5	10	10	0	0.20	0.20	0.00	1	0	11.52	1.40	4	0	10	10	0.00	0.20	0.91	0	1	1.23	11.29
4	0	10	0	0.00	0.20	0.00	1	0	11.29	1.23	3	0	0	10	0.00	0.00	0.91	0	1	1.09	11.10
5	10	10	0	0.20	0.20	0.00	1	0	11.52	1.40	5	10	20	0	0.20	0.39	0.00	1	0	11.61	1.48
5	10	20	0	0.20	0.39	0.00	1	0	11.61	1.48	4	0	20	0	0.00	0.39	0.00	1	0	11.36	1.29
4	0	20	0	0.00	0.39	0.00	1	0	11.36	1.29	4	0	10	0	0.00	0.20	0.00	1	0	11.29	1.23
5	10	10	10	0.20	0.20	0.91	0	1	1.40	11.52	5	10	20	10	0.20	0.39	0.91	0	1	1.48	11.61
5	10	20	10	0.20	0.39	0.91	0	1	1.48	11.61	4	0	20	10	0.00	0.39	0.91	0	1	1.29	11.36
4	0	20	10	0.00	0.39	0.91	0	1	1.29	11.36	4	0	10	10	0.00	0.20	0.91	0	1	1.23	11.29
5	10	20	0	0.20	0.39	0.00	1	0	11.61	1.48	5	10	20	10	0.20	0.39	0.91	0	1	1.48	11.61
4	0	20	0	0.00	0.39	0.00	1	0	11.36	1.29	4	0	20	10	0.00	0.39	0.91	0	1	1.29	11.36
4	0	10	0	0.00	0.20	0.00	1	0	11.29	1.23	5	10	10	10	0.20	0.20	0.91	0	1	1.40	11.52
5	10	10	0	0.20	0.20	0.00	1	0	11.52	1.40	5	10	20	10	0.20	0.39	0.91	0	1	1.48	11.61
5	10	20	0	0.20	0.39	0.00	1	0	11.61	1.48	4	0	20	10	0.00	0.39	0.91	0	1	1.29	11.36
4	0	20	0	0.00	0.39	0.00	1	0	11.36	1.29	4	0	10	10	0.00	0.20	0.91	0	1	1.23	11.29
5	10	20	0	0.20	0.39	0.00	1	0	11.61	1.48	5	10	30	0	0.20	0.59	0.00	1	0	11.61	1.48
5	10	30	0	0.20	0.59	0.00	1	0	11.61	1.48	4	0	30	0	0.00	0.59	0.00	1	0	11.36	1.29
4	0	30	0	0.00	0.59	0.00	1	0	11.36	1.29	4	0	20	0	0.00	0.39	0.00	1	0	11.36	1.29
5	10	20	10	0.20	0.39	0.91	0	1	1.48	11.61	5	10	30	10	0.20	0.59	0.91	0	1	1.48	11.61
5	10	30	10	0.20	0.59	0.91	0	1	1.48	11.61	4	0	30	10	0.00	0.59	0.91	0	1	1.29	11.36
4	0	30	10	0.00	0.59	0.91	0	1	1.29	11.36	4	0	20	10	0.00	0.39	0.91	0	1	1.29	11.36
5	10	30	0	0.20	0.59	0.00	1	0	11.61	1.48	5	10	30	10	0.20	0.59	0.91	0	1	1.48	11.61

Figure 21: Feature vectors of the 361-member slab

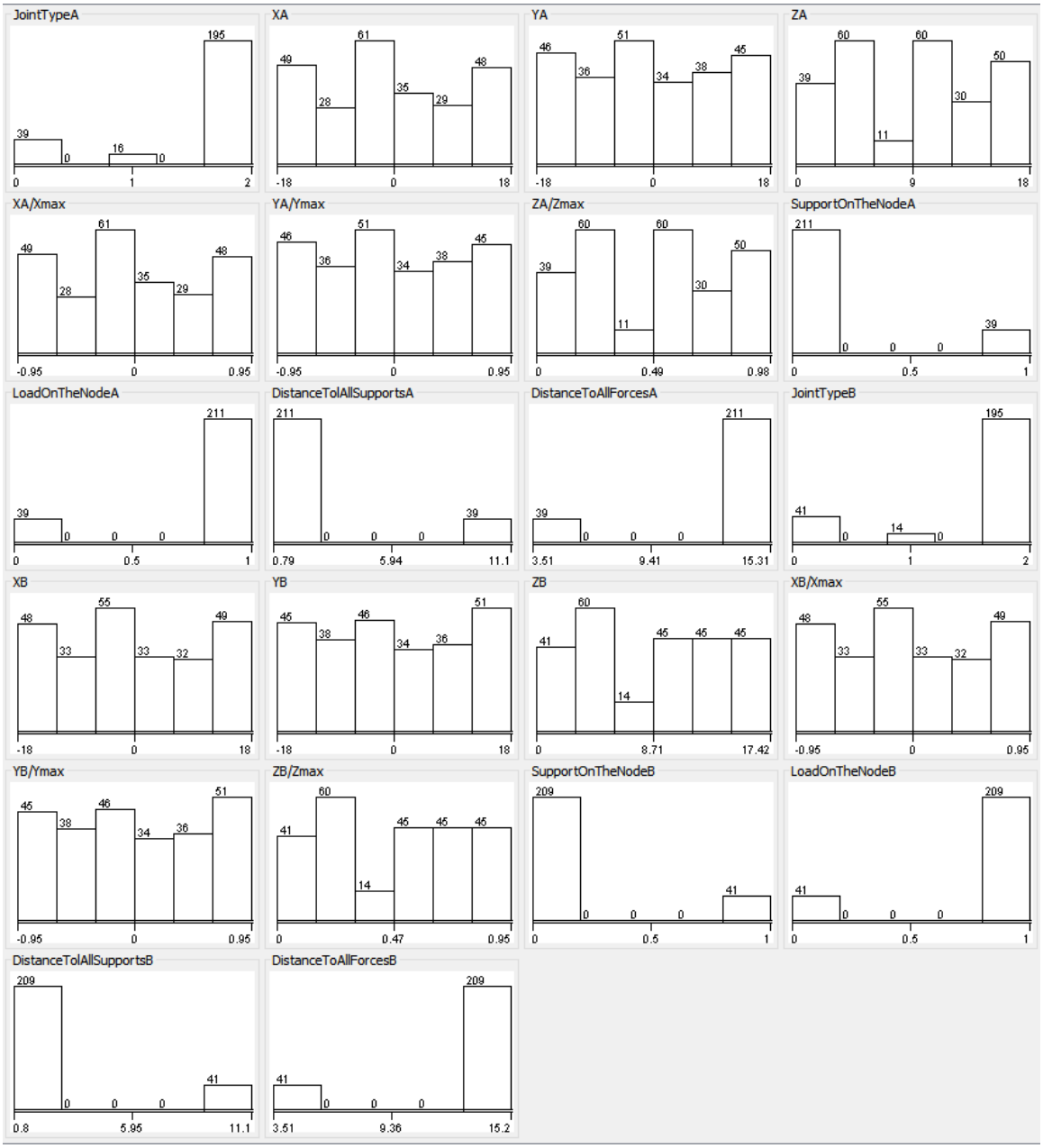


Figure 22: Visualization of the features of the 250-member dome

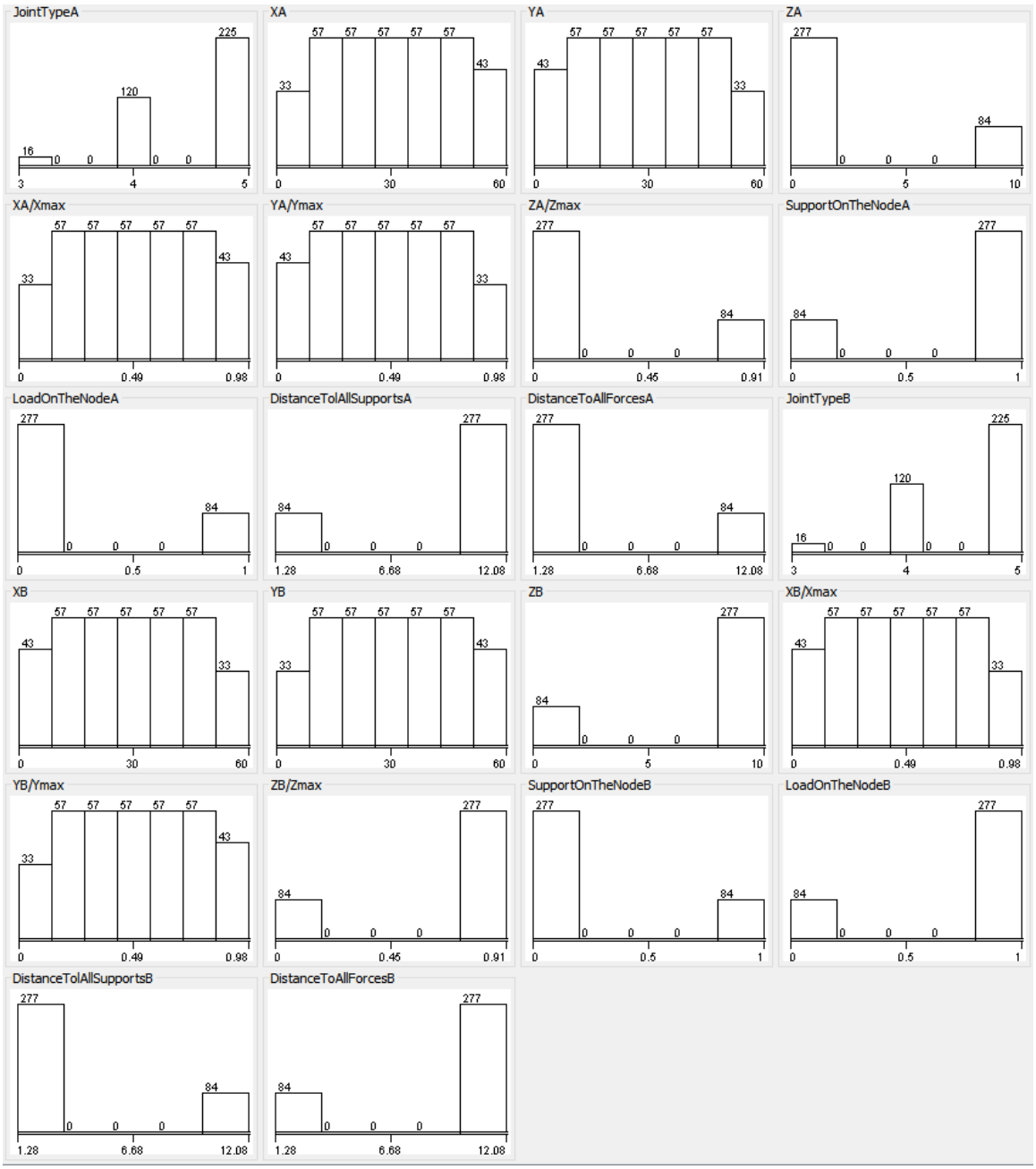


Figure 23: Visualization of the features of the 361-member slab

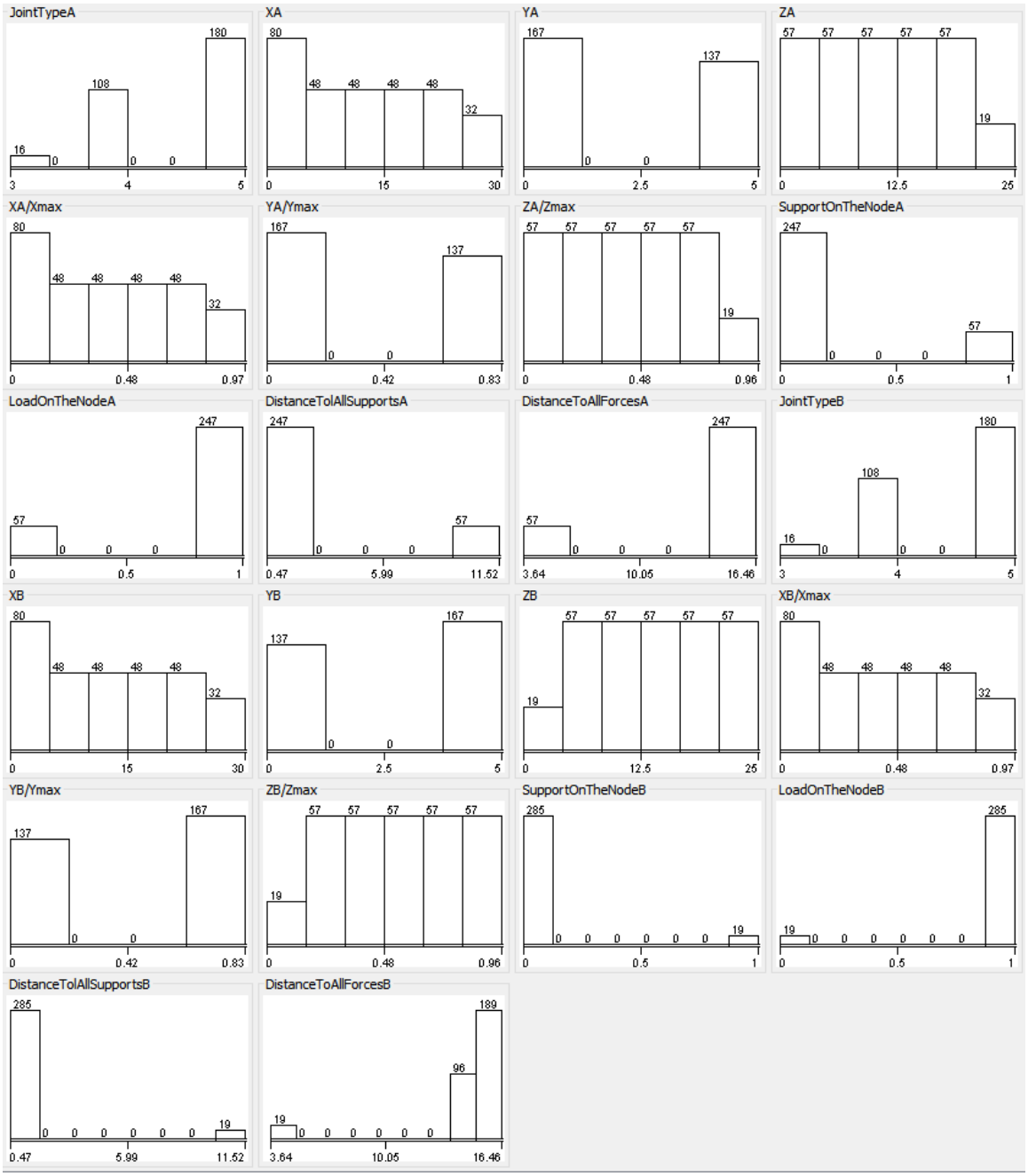


Figure 24: Visualization of the features of the 304-member wall

Step 2: Structural Analysis

For structural analysis, the coordination of nodes, the connectivity of edges, and the location of forces and supports were imported from Dynamo-generated text files into Autodesk Nastran. For each edge, a cross-sectional area between 0.0001 and 0.01 m^2 (0.155-15.5 in^2) was randomly selected and added to the feature vectors. In addition, the

local stiffness of each member was computed based on the stiffness of the neighborhood of the member. Figure 25 demonstrates the local stiffness of the structure for the red member. The green members are connected to the left node, and the orange members are connected to the right node. The black members are connected to the green and orange members. The sum of the stiffness of all of these members are calculated and added to the feature vectors.

Table 5: List of all features for members

Features	Description	Example
Member Area	Cross-sectional area of member	0.001
No of Neighbors	No. of members in a neighborhood (neighbors and neighbors of neighbors)	41
Local Stiffness	Stiffness of the neighborhood (neighbors and neighbors of neighbors)	10.23

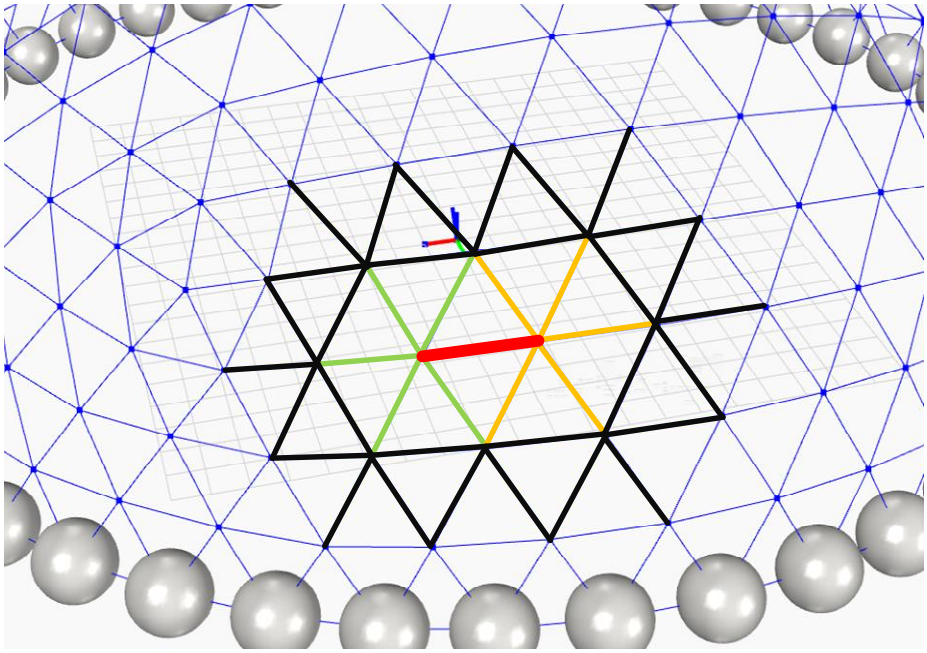


Figure 25: Local stiffness of a member

Each edge was modeled using the CBEAM element in Nastran for fixed-joint structures and the CBAR element for pinned-joint structures. For both structures, the translation and rotation of all of the nodes at the ground level are constrained. In addition, a constant force of 30,000 N was applied to the structure as follows.

- **Dome and Slab:** Force in the -z direction on all nodes that are not on the ground
- **Wall:** Force in the -z direction on all nodes at the highest level (see Figure 26)

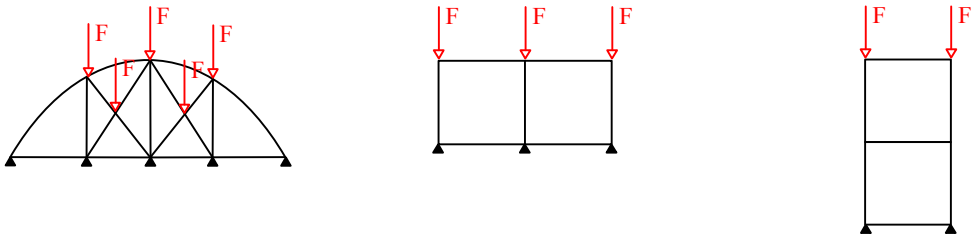


Figure 26: Simplified 2D-representation of forces applied to domes, slabs, and walls

The material of the structure is steel with Young's module of 210 GPA, a density of 7,800 kg/m3, and Poisson's ratio of 0.26. Two outputs of the analysis, min/max stress of each bar in pinned-joint and Von Mises Stress in fixed-joint structures, are added to the feature list.

Table 6: Labels of feature vectors

Features	Description	Example
Stress	Stress value of each member	1.2E6

Phase 2: Model Creation and Verification

This study uses a deep neural network library known as Theano for defining, optimizing, and evaluating mathematic expressions (Bastien, Lamblin et al. 2012). This library integrates NumPy and SciPy for matrix calculation, generates dynamic C code to

more rapidly evaluate expressions, and uses GPU, which makes it 140X faster than a CPU (Theano 2015).

Theano uses a symbolic representation of variables. For instance, x in x = Theano.dscalar('x') is a symbol or variable object that is of the Theano.dscalar type. This type assigns a 0-dimensional array of doubles to the variable x. The following Theano expression shows how to add two scalar numbers. The result of running this code is a scalar number of five.

```
import theano.tensor as T
from theano import *
a= T.scalar('a')
b=T.scalar('b')
c=a+b
f = function( [a,b],c )
f(2,3)
```

Figure 27: An example of a Theano expression

To find the most suitable architecture of a neural network, this study uses various network architectures. These networks are going to be tested against the build time and accuracy so that users can choose a network architecture suitable to their problem. For a one-hidden-layer neural network model in regression problems, we use sigmoid as an activation function on the hidden later and a tangent hyperbolic function on the output layer. However, the output layer does not have an activation function for two main reasons. First, all data are standardized to fit between -1 and 1. Thus, the predicted values are expected to be in this range. Second, from a structural optimization perspective, if the predicted values are not in this range, the outcome of the evaluation is a failure, meaning that the structure cannot tolerate the load. Hence, all numbers more than 1 or less than -1 can be set to 1 or -1, respectively.

```
def regression_model(X, w_h, w_o):
    h = T.nnet.sigmoid(T.dot(X, w_h))
    pyx = (T.dot(h, w_o))
    return pyx
```

Figure 28: Regression model

The creation and verification of the neural network models of this study is presented in Figure 29. The description of the process is as follows.

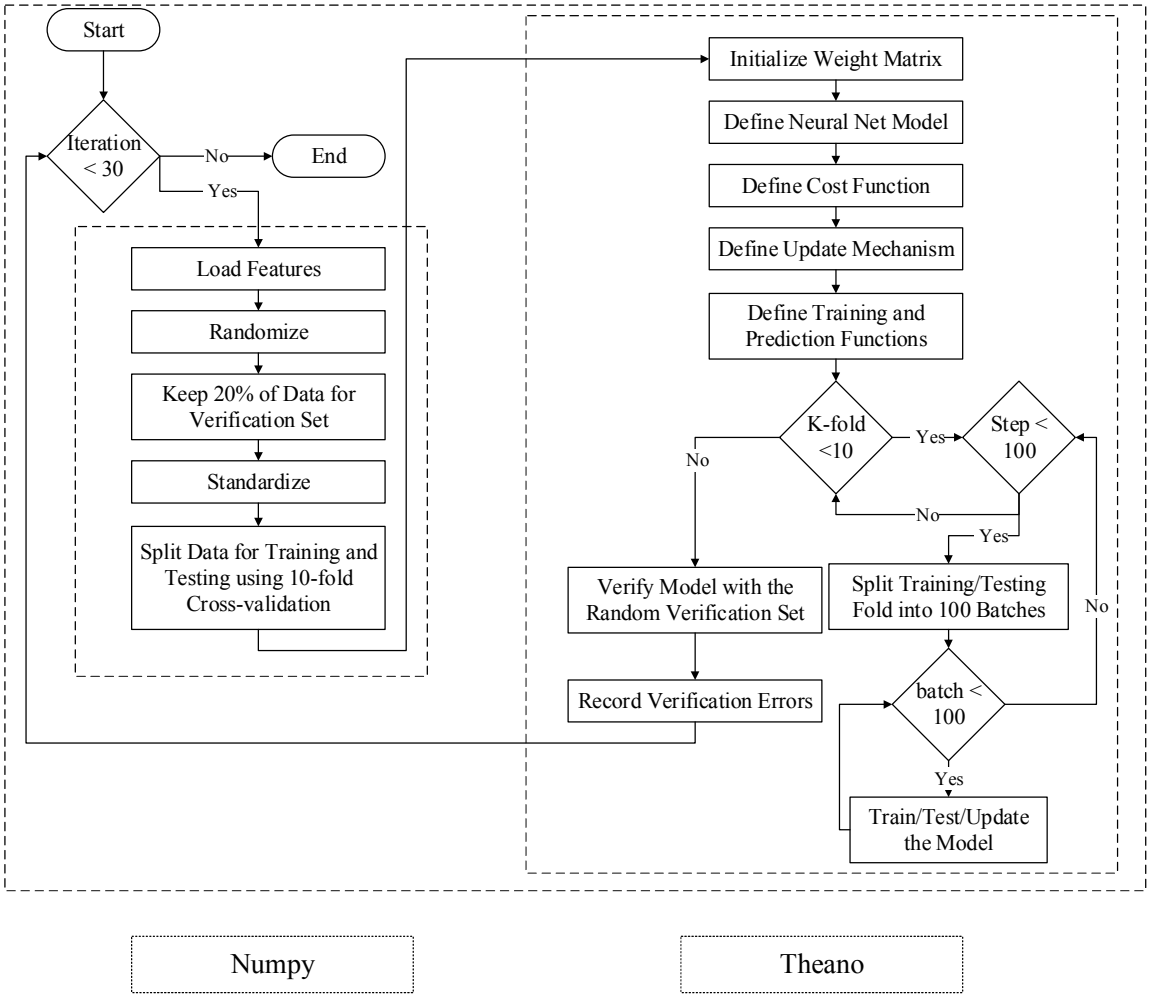


Figure 29: Model creation and verification

a) Load Features

The first step in the model creation phase is to load sample data (features) into NumPy arrays. The number of samples varies from 1,000 to 480,000, based on the hypothesis.

b) Randomize Samples

These samples are being randomized to ensure that the model does not depend on the arrangement or the order of input data.

c) Retain Data for the Verification Set

After loading the data, 20% of the samples are kept for verifying the predictability of the model. Using the verification set, we examine the performance of the neural net model after completing the training to assure that the model is not over- or under-fit.

d) Standardize

The range of each of the 26 features differs from that of the other features. For instance, the range of the design type is from 1 to 6, and the area is from 0.001 to 0.1. In this case, during the training of the neural network model, the contribution of the design type would be much greater than that of the area. Therefore, it is critical to scale input data such that all of the features have the same range. In this study, all feature vectors are rescaled to a midrange of zero and a range of two.

e) Cross-validate

A validation technique for estimating the generalization error of statistical models is called cross-validation. In a k-fold cross-validation technique, input samples are divided into k subsets and the neural net model is trained k times. In each training, we leave one subset out for validating the performance of the model and train the model with other subsets (Weiss and Kulikowski 1991).

f) Initialize the Weight Matrix

At the beginning of the training, the weight matrix of the model is initialized with random numbers. These numbers will be adjusted during the training process so as to minimize the mean-square error of the model.

g) Calculate the Cost Function

The cost function of the neural network models of this study is the square error between the predicted and actual values of the model (see Equation 8).

h) Update the Weight Matrix

After initializing the weight, a back propagation algorithm uses all training data to find the weight of neurons with the least amount of error. In each iteration of learning, these weights are updated and recorded for the next iterations.

i) Train and Predict Functions

In this study, the input of the training function is a matrix of the input and output features. The output of the training function is the cost function (i.e., mean-square error between the predicted and actual values). By contrast, the input of the prediction function is only a matrix of the input features and its output is the predicted stress values.

j) Split Training/Testing Fold into 100 Batches

In this step, the input and output matrices are divided into 100 batches. Then by iterating over all batches and maintaining the weight of the model constant, the Delta values of the weight and the bias and the errors of the model are computed. Finally, the aggregated Delta values are used to update the weight and the bias of the model. This process is repeated 100 times.

CHAPTER 4

TEST AND VALIDATION OF GENERALIZABLE METAMODELS

This chapter demonstrates testing and validation of generalizable metamodels using four hypotheses. For each hypothesis, the experiment setup, methodology, results, and discussion is presented.

Hypothesis 1

Ha: The architecture and internal weights of the neural network models of 3D-trusses and space frames differ significantly.

H0: The architecture and internal weights of the neural network models of 3D-trusses and space frames do not differ significantly.

Experiment Setup

Geometry

This experiment tests a 250-member dome with the radius of 18 m (59.06 ft.). The dome has 91 nodes, 20 of which connect the structure to the ground and 71 of which carry a constant load of 30,000N.

Features

The input of the model is the cross-sectional area of all 250 members, and the output is the stress values of all members.

Sampling Plan

Using Autodesk Nastran, we generate 1,000 samples. In each sample, the cross-sectional areas of members were randomly chosen between 0.1 and 0.001 m^2, and the stress values of members were computed. Samples of the fixed- and pinned-joint structures were generated using CBEAM and CROD elements, respectively.

Methodology

Implementing the experiment involves two main steps: preparing the sample data and creating the neural net model (Figure 30). In the data preparation step, all of the sample data are loaded into a NumPy array. After randomization and standardization, the data are split into training (60%), testing (20%), and verification (20%) sets. The neural net model in this experiment has one hidden layer, with 250 neurons for inputs and outputs. The cost function in this experiment is the mean square error of actual versus predicted stress values. By changing the number of neurons in the hidden layer from 10 to 155, the change in the root mean square error of the model is recorded. In addition to test the null hypothesis, we compare the internal weights of a pinned- and fixed-joint model with 70 neurons using the student t-test.

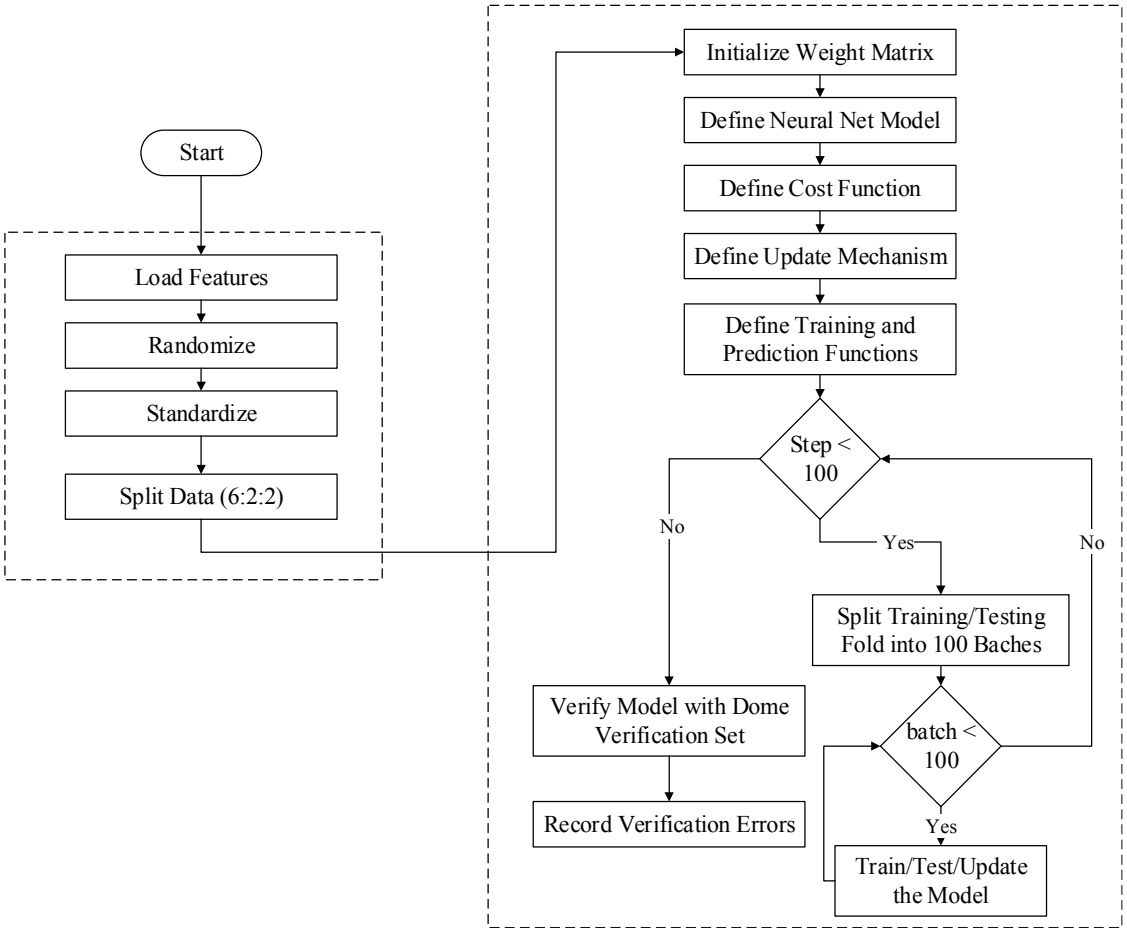


Figure 30: Implementation of Hypothesis 1

Results

The results show that the fixed-joint neural net model has a higher root mean square error than the pinned-joint truss model. One explanation for this finding is that the impact of shear and the moment of intertie in fixed-joint structures make the prediction of the structural performance more difficult. In addition, the number of hidden neurons required in the pinned-joint model is slightly less than that in the fixed-joint model. The results of the student t-test reveal a significant difference between the two input matrices at the hidden layer; however, the results showed no significant difference between the matrices of output layer.

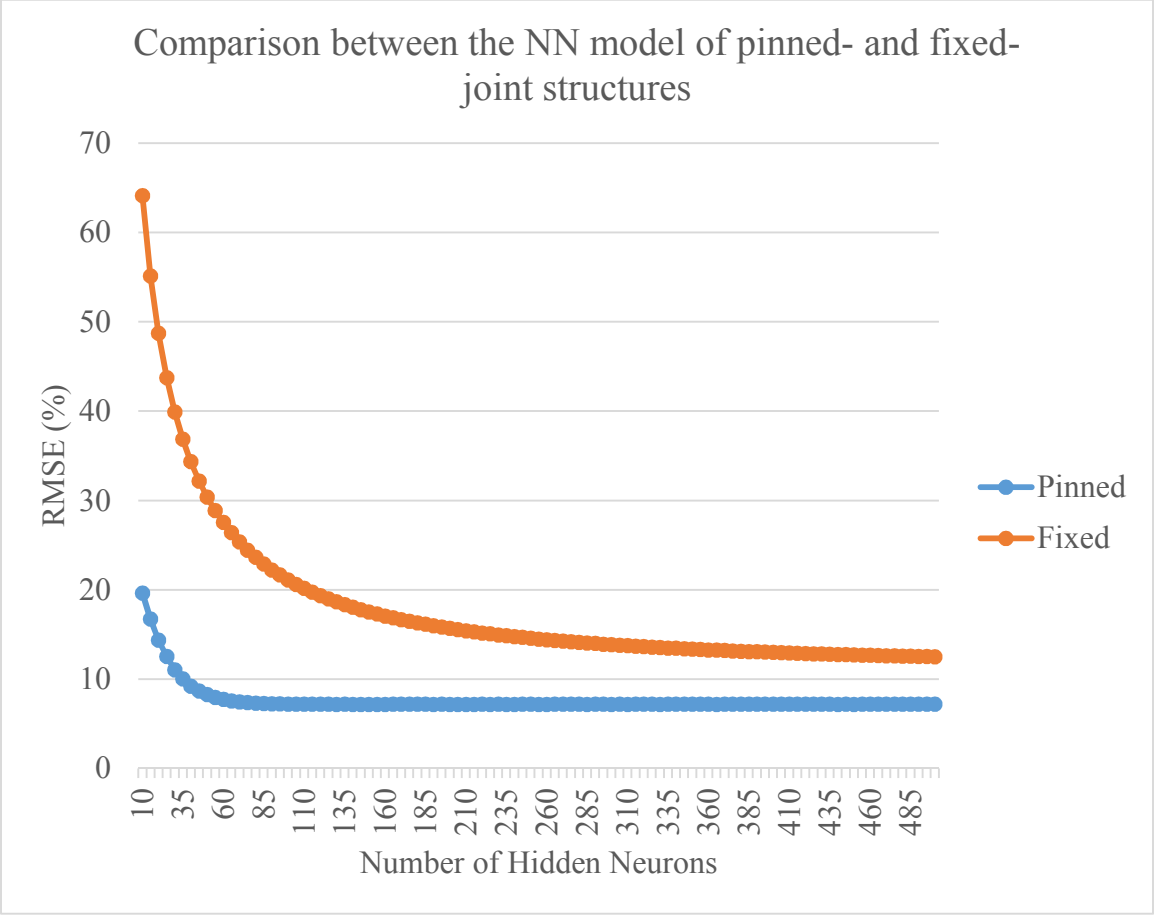


Figure 31: Comparison between the neural network models of the pinned- and fixed-joint structures

Table 7: Results of the t-test for pinned- and fixed joint structures

	Hidden Layer		Output Layer	
	Fixed	Pinned	Fixed	Pinned
Mean	6.41E-06	4.85E-06	-3.96E-03	7.39E-04
SD	0.010	0.010	0.010	0.010
P	0.962		0.000	

Discussion

The results of this hypothesis showed a significant difference between the internal weights of pinned- and fixed-joint structures. In addition, it found that the error rate of the model in fixed-joint structures is slightly higher than that of pinned-joint structures. However, both structures exhibited the same rate of errors because of the similarity in the slope of the errors with a certain number of neurons (Figure 31). Therefore, even though the models are statistically different, they are, in practice, similar in terms of the mechanics of errors.

Hypothesis 2

A generalizable metamodel can more accurately predict the structural performance of space frames than exclusive metamodels.

Experiment Setup

Case 1-Exclusive Neural Net model

For the exclusive neural net model, we use 1,000 samples for the fixed-joint 250-member dome used to test Hypothesis 1.

Case 2- Generalizable Neural Net model

Features

To encode the property of structural members to machine-readable numerical values in generalizable neural nets, this study proposes 14 unique features, 11 of which are node descriptors and three-member descriptor. Hence, a structural member with two nodes (left and right node) has 25 (11*2+3) features.

Sampling Plan

We use the same 1,000 samples in Case 1, which is converted to 250,000 samples using the feature descriptor proposed in this study. To assure a correct comparison between two cases (generalizable and exclusive), we use the same sets of training, testing, and verification without randomization. In other words, the size of the members and the stress values of both cases are the same.

Methodology

The generalizable neural net model has 25 inputs (descriptors) and one output (stress value). The number of hidden neurons changed from five to 100, and the root mean squared errors of both models were calculated. All parameters of the model remained the same in both exclusive and generalizable neural net models.

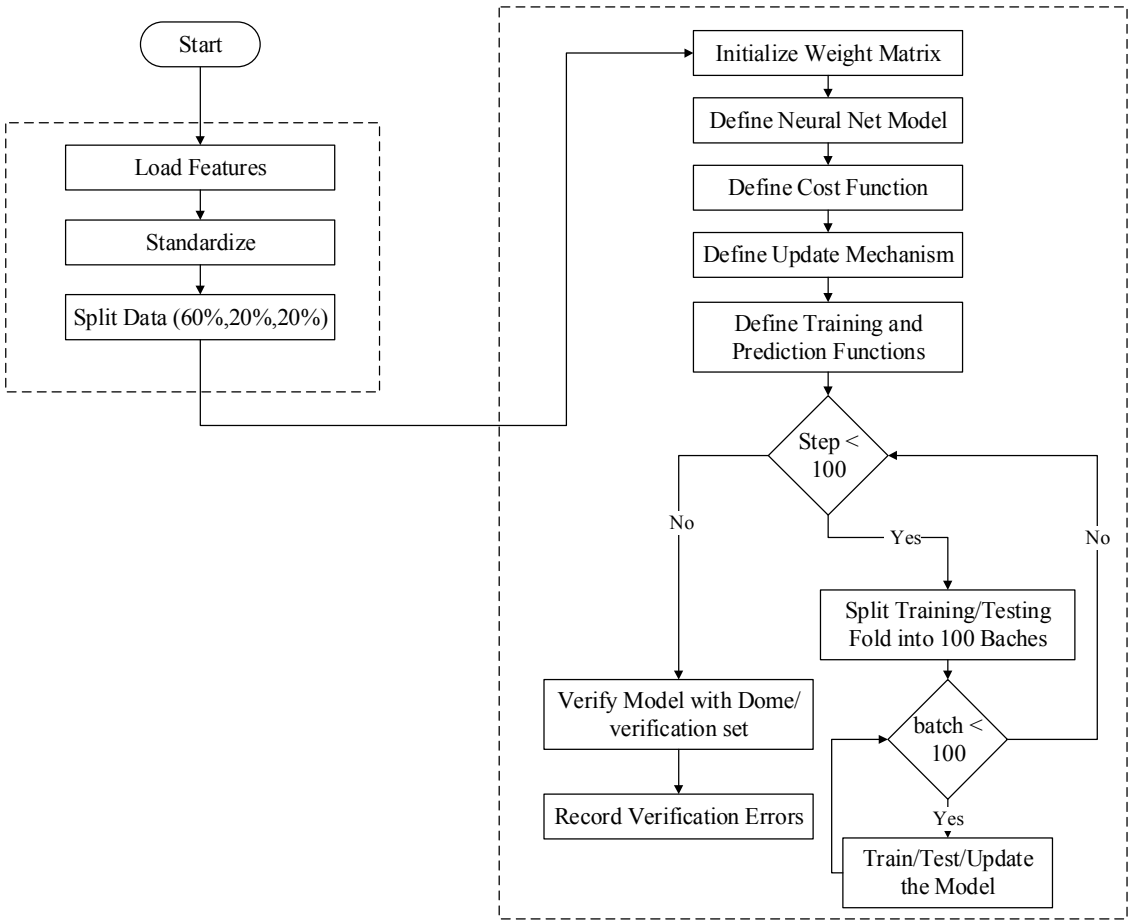


Figure 32: Implementation of Hypothesis 2

Results

The results show that the exclusive neural net has a high error rate with few neurons in the hidden layer. As the number of neurons increases, the error rate decreases and stabilizes at 5%. The performance of the model does not improve if it has more than 70 neurons in the hidden layer. By contrast, the generalizable neural net exhibits a steady percentage of errors of 5.6% if the number of neurons changes from five to 80. Adding more than 80 neurons to the network dramatically increases the number of errors (Table 8 and Figure 33).

Table 8: Comparison between the error rate of the exclusive and generalizable neural networks

No. of Neurons	5	10	15	20	25	30	35	40	45	50
RMSE Exclusive	11.2	9.74	8.45	7.53	6.86	6.34	5.96	5.69	5.46	5.37
RMSE Generalizable	5.64	5.64	5.64	5.64	5.64	5.64	5.64	5.64	5.64	5.64
No. of Neurons	55	60	65	70	75	80	85	90	95	100
RMSE Exclusive	5.27	5.21	5.17	5.14	5.11	5.10	5.09	5.09	5.08	5.08
RMSE Generalizable	5.64	5.64	5.64	5.64	5.64	5.64	9.14	13.51	16.46	19.47

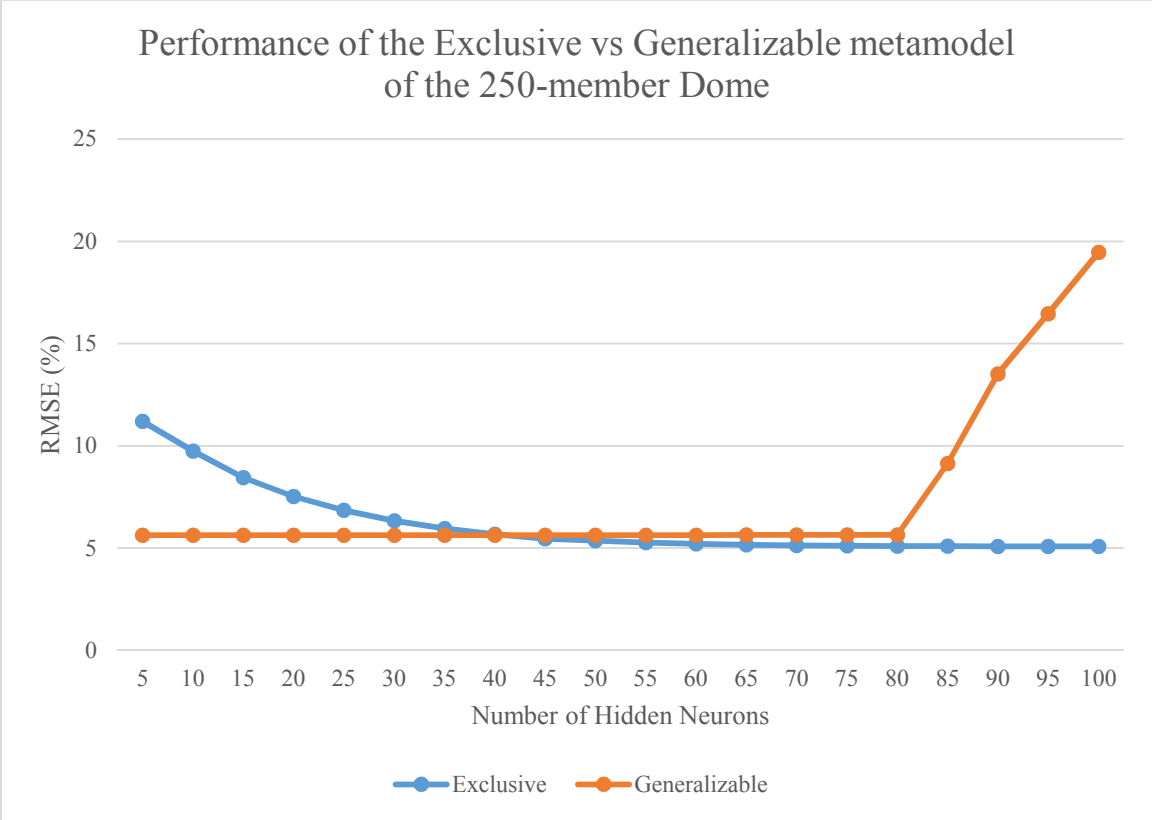


Figure 33: Performance of the exclusive and generalizable metamodels of the 250-member dome

Discussion

Generalizable metamodels can predict more accurate or as well as exclusive metamodels until the number of neurons reaches to the threshold of the model. Even though the measure of errors in exclusive neural net models was shown to be better than that of generalizable neural net models, when the number of neurons are between 40 and 80, the difference between the errors of models is negligible (0.6 %). It is worth noting that the exclusive neural net is capable of predicting the structural performance of only one structure. By contrast, a generalizable model is designed to predict the performance of a variety of structures.

Hypothesis 3

The results of the finite element analysis of various structures can be used for creating a metamodel that predicts the stress values of the members of these structures in size optimization problems.

Experiment Setup

Case 1- Data Combination in the Same Class of Structures

In this study, the term "data combination" or "data aggregation" means mixing the sample of analysis data from various structures. For instance, the first five rows of the Figure 34 are the feature vectors of a 250-member dome and the next five rows are the feature vectors of a 700-member dome.

	Joint Type	X node	Y node	Z node	X / X max	Y / Y max	Z / Z max	Support?	Load?	Support dist	Load dist	Joint Type	X node	Y node	Z node	X / X max	Y / Y max	Z / Z max	Support?	Load?	Support dist	Load dist
Dome 250	3	0	0	0	0.00	0.00	0.00	1	0	11.10	1.09	4	10	0	0	0.20	0.00	0.00	1	0	11.29	1.23
	4	10	0	0	0.20	0.00	0.00	1	0	11.29	1.23	5	10	10	0	0.20	0.20	0.00	1	0	11.52	1.40
	5	10	10	0	0.20	0.20	0.00	1	0	11.52	1.40	4	0	10	0	0.00	0.20	0.00	1	0	11.29	1.23
	4	0	10	0	0.00	0.20	0.00	1	0	11.29	1.23	3	0	0	0	0.00	0.00	0.00	1	0	11.10	1.09
	3	0	0	10	0.00	0.00	0.91	0	1	1.09	11.10	4	10	0	10	0.20	0.00	0.91	0	1	1.23	11.29
Dome 700	4	10	0	10	0.20	0.00	0.91	0	1	1.23	11.29	5	10	10	10	0.20	0.20	0.91	0	1	1.40	11.52
	5	10	10	10	0.20	0.20	0.91	0	1	1.40	11.52	4	0	10	10	0.00	0.20	0.91	0	1	1.23	11.29
	4	0	10	10	0.00	0.20	0.91	0	1	1.23	11.29	3	0	0	10	0.00	0.00	0.91	0	1	1.09	11.10
	3	0	0	0	0.00	0.00	0.00	1	0	11.10	1.09	3	0	0	10	0.00	0.00	0.91	0	1	1.09	11.10
	4	10	0	0	0.20	0.00	0.00	1	0	11.29	1.23	4	10	0	10	0.20	0.00	0.91	0	1	1.23	11.29
	5	10	10	0	0.20	0.20	0.00	1	0	11.52	1.40	5	10	10	10	0.20	0.20	0.91	0	1	1.40	11.52
	4	0	10	0	0.00	0.20	0.00	1	0	11.29	1.23	4	0	10	10	0.00	0.20	0.91	0	1	1.23	11.29

Figure 34: Data combination

Geometry

This experiment generates and tests four variations of a dome, a slab, and a wall.

Features

This experiment consists of 11 node features and three member features. The length of the input vector is 25, and the length of the output vector is one.

Sampling Plan

Each geometry entails the use of three sampling plans (Table 9-12). The first plan comprises training models with a batch of 40,000 samples without combining the data. The second plan combines data into a data pool and retains the same number of samples (i.e., 40,000) in each batch. For instance, 20,000 data from the 250-member dome (D250) are added to 20,000 samples from the 700-member dome (D700). The third plan combines samples with a fixed length (40,000) without removing any samples, creating a variable batch size.

Table 9: Sampling plan for a dome

Plan 1: Individual(40k_Batch)	Plan 2: Data Pool (40k_ Batch)
D250 (40,000)	D250 (40,000)
D700 (40,000)	D250 (20,000) D700 (20,000)
D980 (40,000)	D250 (13,333) D700 (13,333) D980 (13,334)
D1525 (40,000)	D250 (10,000) D700 (10,000) D980 (10,000) D1525 (10,000)

Table 10: Sampling plan for a slab

Plan 1: Individual (40k_Batch)	Plan 2: Data Pool (40k_ Batch)
S361 (40,000)	S361 (40,000)
S784 (40,000)	S361 (20,000) S784 (20,000)
S961 (40,000)	S361 (13,333) S784 (13,333) S961 (13,334)
S1600 (40,000)	S361 (10,000) S784 (10,000) S961 (10,000) S1600 (10,000)

Table 11: Sampling plan for a wall

Plan 1: Individual (40k_Batch)	Plan 2: Data Pool (40k_ Batch)
W304 (40,000)	W304 (40,000)
W736 (40,000)	W304 (20,000) W736 (20,000)
W928 (40,000)	W304 (13,333) W736 (13,333) W928 (13,334)
W1504 (40,000)	W304 (10,000) W736 (10,000) W928 (10,000) W1504 (10,000)

Methodology

These data are loaded into a NumPy array, randomized, and standardized. This array is split into an 80:20 ratio for training/testing and verifying the neural net models. The training/testing set is partitioned into ten sections for a ten-fold cross-validation. In each fold, two partitions of the data are assigned to the testing set, and eight other partitions are added to the training set. After completing the training of the model, we compute and record the performance of the model using the verification set. The entire process of training, from loading data to recording the measure of errors, is repeated 30 times to ensure that the results are neither random nor biased. The average number of errors of these 30 iterations is reported as the performance metric of the model.

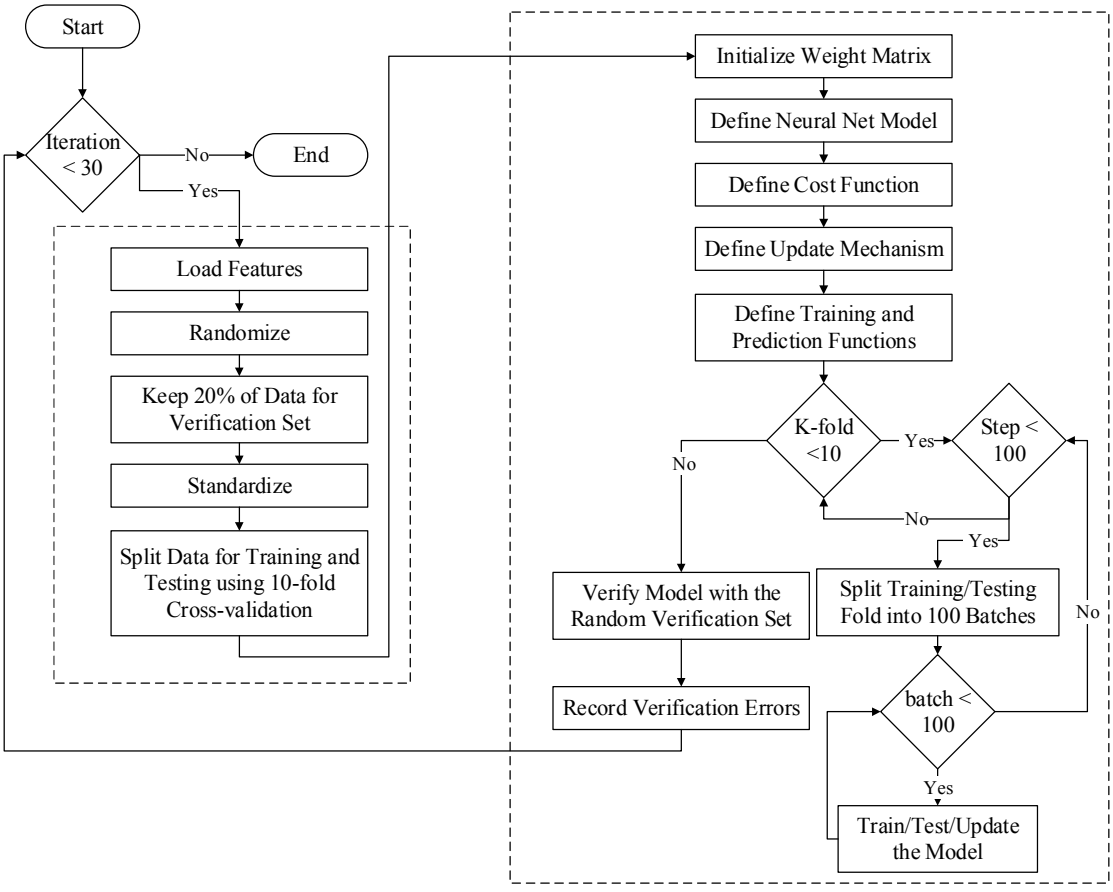


Figure 35: Implementation of Hypothesis 3

Results

A comparison between the individual domes and the data pool shows that models generated from the combination of samples have lower errors compared to the models generated from samples of individual domes (Figure 36). For instance, the number of errors of the model generated from the samples of the 1,525-member dome is almost nine times as many as that of the combination of the samples of the 250-, 700-, 980-, and 1525-member domes.

Similarly, a comparison between the individual slabs and walls and the data pool shows that the models generated from the combination of data generate fewer errors (Figure 37-38).

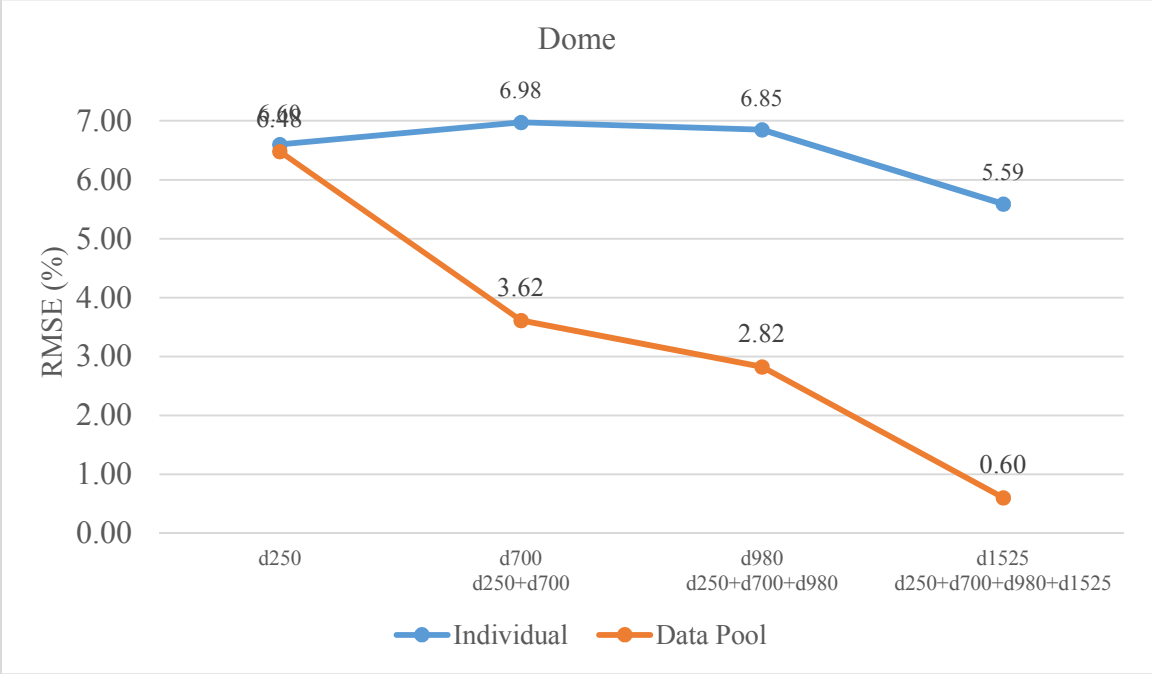


Figure 36: Comparison between the models of the data pool and those of individual samples of domes

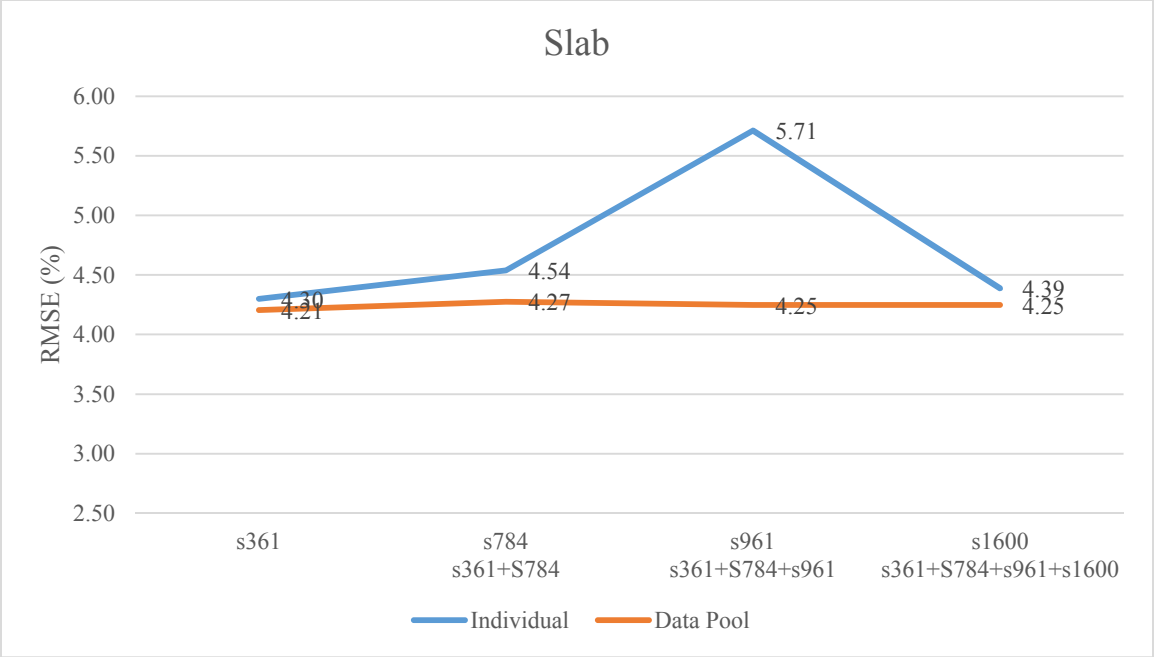


Figure 37: Comparison between the models of the data pool and those of individual samples of slabs

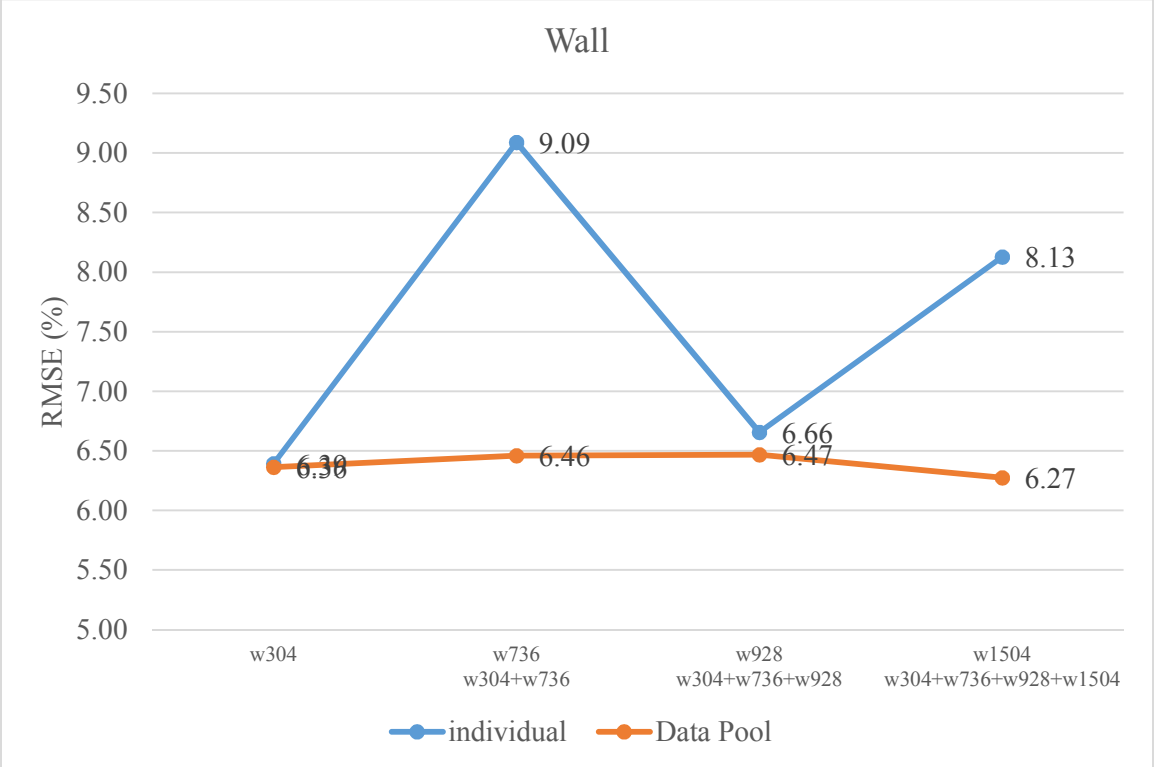


Figure 38: Comparison between the models of the data pool and those of the individual samples of walls

Case 2- Data Combination of Various Structures

Features

All of the features of Case 1 were used in Case 2.

Sampling Plan

The only difference between Cases 1 and 2 is the sampling strategy. Two main plans in Case 2 were combining all of the sample data and maintaining a fixed size of 40,000 versus having 40,000 data points for each structure (480,000 samples).

Table 12: Sampling plan for the data pool (DSW1-2)

DSW1: Data Pool (40k_Batch)	DSW1: Data Pool (40,000_Each)	DSW2: Data Pool (40k_Batch)	DSW2: Data Pool (40,000_Each)
D250 (13,333)	D250 (40,000)	D250 (6,666)	D250 (40,000)
		D700 (6,667)	D700 (40,000)
S361 (13,333)	S361 (40,000)		
		S361 (6,666)	S361 (40,000)
W304 (13,334)	W304 (40,000)	S784 (6,667)	S784 (40,000)
		W304 (6,667)	W304 (40,000)
		W736 (6,667)	W736 (40,000)

Table 13: Sampling plan for the data pool (DSW3-4)

DSW3: Data Pool (40k_Batch)	DSW3: Data Pool (40,000_Each)	DSW4: Data Pool (40k_Batch)	DSW4: Data Pool (40,000_Each)
D250 (4,444)	D250 (40,000)	D250 (3,333)	D250 (40,000)
D700 (4,444)	D700 (40,000)	D700 (3,333)	D700 (40,000)
D980 (4,445)	D980 (40,000)	D980 (3,333)	D980 (40,000)
		D1525 (3,334)	D1525 (40,000)
S361 (4,444)	S361 (40,000)		
S784 (4,444)	S784 (40,000)	S361 (3,333)	S361 (40,000)
S961 (4,445)	S961 (40,000)	S784 (3,333)	S784 (40,000)
		S961 (3,333)	S961 (40,000)
W304 (4,444)	W304 (40,000)	S1600 (3,334)	S1600 (40,000)
W736 (4,445)	W736 (40,000)		
W928 (4,445)	W928 (40,000)	W304 (3,333)	W304 (40,000)
		W736 (3,333)	W736 (40,000)
		W928 (3,334)	W928 (40,000)
		W1504 (3,334)	W1504 (40,000)

Results

The results of this experiment shows that the performance of the metamodels improves as the number of samples in the data pool increases. The best model has 40,000 samples from all structures (dsw4) and shows an almost 300% improvement in the number of errors (Figure 39).

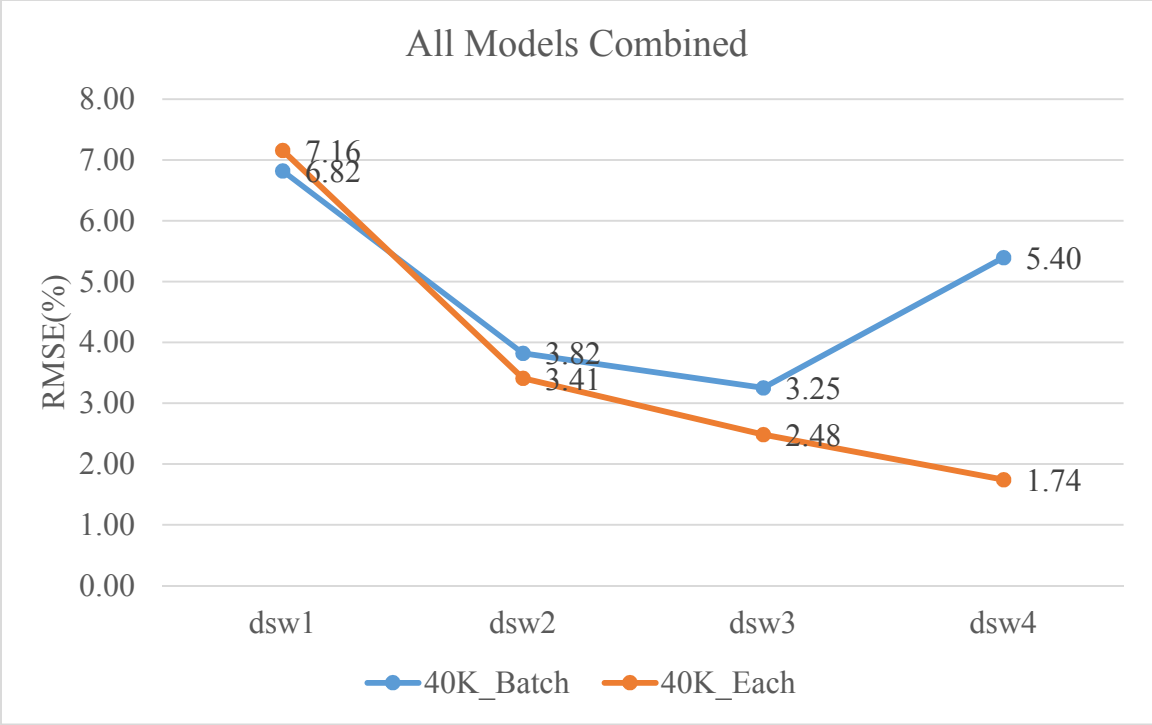


Figure 39: Impact of the number of samples on the performance of models

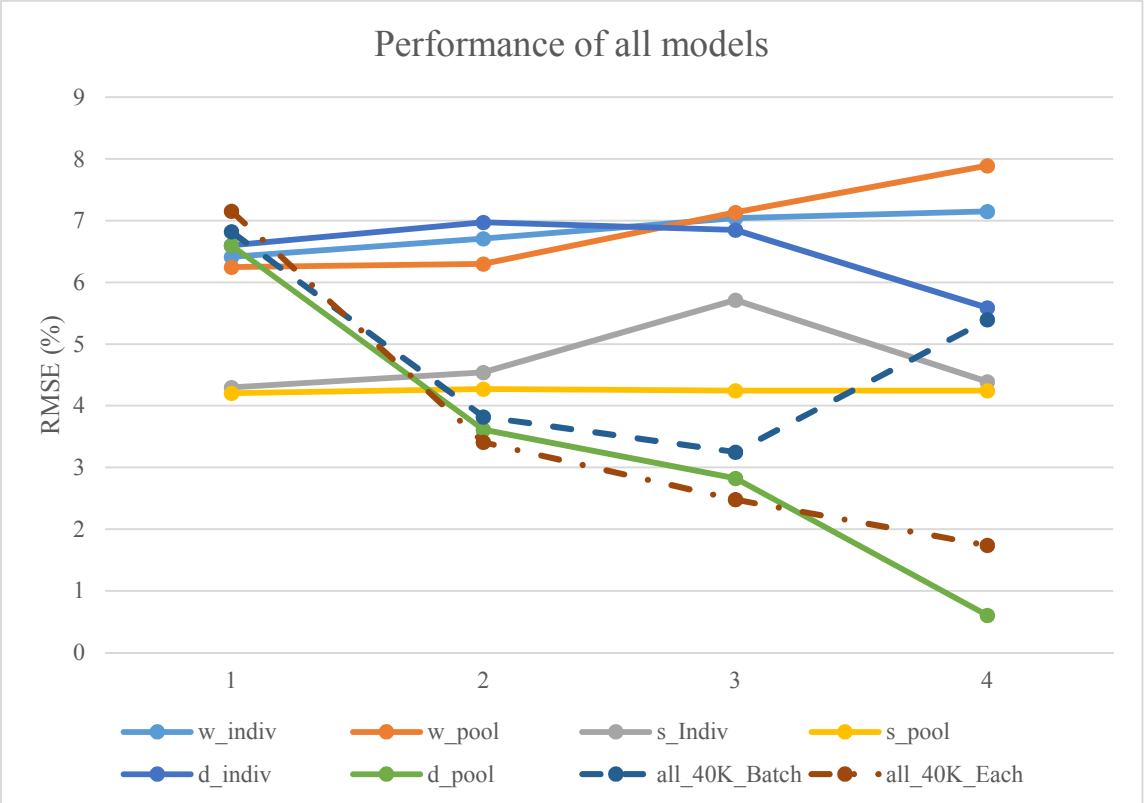


Figure 40: Performance of all models

Discussion

The results of the first experiment showed that metamodels generated from the combination of data in each class of geometry produced fewer errors than the metamodels generated from individual geometries. For instance, the percentage of errors of a model generated from the 736-member wall was 9.09% compared to 6.46% from the combination of the 304- and 736-member walls.

The results of the second experiment demonstrated that the finite element analysis of various structures could be used for creating metamodels that infer the stress values of the members of these structures. The performance of these generalizable metamodels lay somewhere between that of all metamodels generated from the samples of individual structures (Figure 40). In this experiment, the trend of the generalizable metamodels that included sample data from all of the structures (dashed lines in Figure 40) was closer to that of the dome.

Hypothesis 4

The results of the finite element analysis of various structures can be used for creating a metamodel that predicts the performance of structures within the same "geometry class" in size optimization problems.

Experiment Setup

Geometry

This experiment uses all four variations of the three structures.

Features

The experiment entails the use of 11 node features and three member features.

Sampling Plan

Case 1: Individual Structures

In this case, for each structure type, we maintain the structure with a minimum number of members for the verification set and build and test the neural net models using data from other structures. For example, we retain the dome with 250 members for prediction and a combination of samples from other domes (D700, D980, and D1525) for training the model. To assure that all samples have a sufficient number of complete structures, the experiment entails no randomization after loading the data. For instance, 10,000 samples of the 250-member dome comprise 40 complete domes.

Case 2: Combination of Structures

In this case, we combine the data from the dome, the slab, and the wall structures, retaining a size of 40,000 for each sample batch. The verification set is the combination of three 10,000 sets found in Case 1.

Table 14: Sampling plan for Hypothesis 4

	Case 1			Case 2
	Dome (40k_Batch)	Slab (40k_Batch)	Wall (40k_Batch)	Combination (40k_Batch)
Training and Testing	D700 (13,333) D980 (13,333) D1525 (13,334)	S784 (13,333) S961 (13,333) S1600 (13,334)	W736 (13,333) W928 (13,333) W1504 (13,334)	D700 (4,444) D980 (4,444) D1525 (4,445) S784 (4,444) S961 (4,444) S1600 (4,445) W736 (4,444) W928 (4,445) W1504 (4,445)
Verification	D250 (10,000)	S361 (10,000)	W304 (10,000)	D250 (10,000) S361 (10,000) W304 (10,000)

Methodology

The process of training the neural network model in this hypothesis is the same as that of Hypothesis 3. The only difference is that the verification set is pre-defined, not randomly selected. Again, to enhance the quality of the results, we have ten-fold cross-validation that is repeated 30 times.

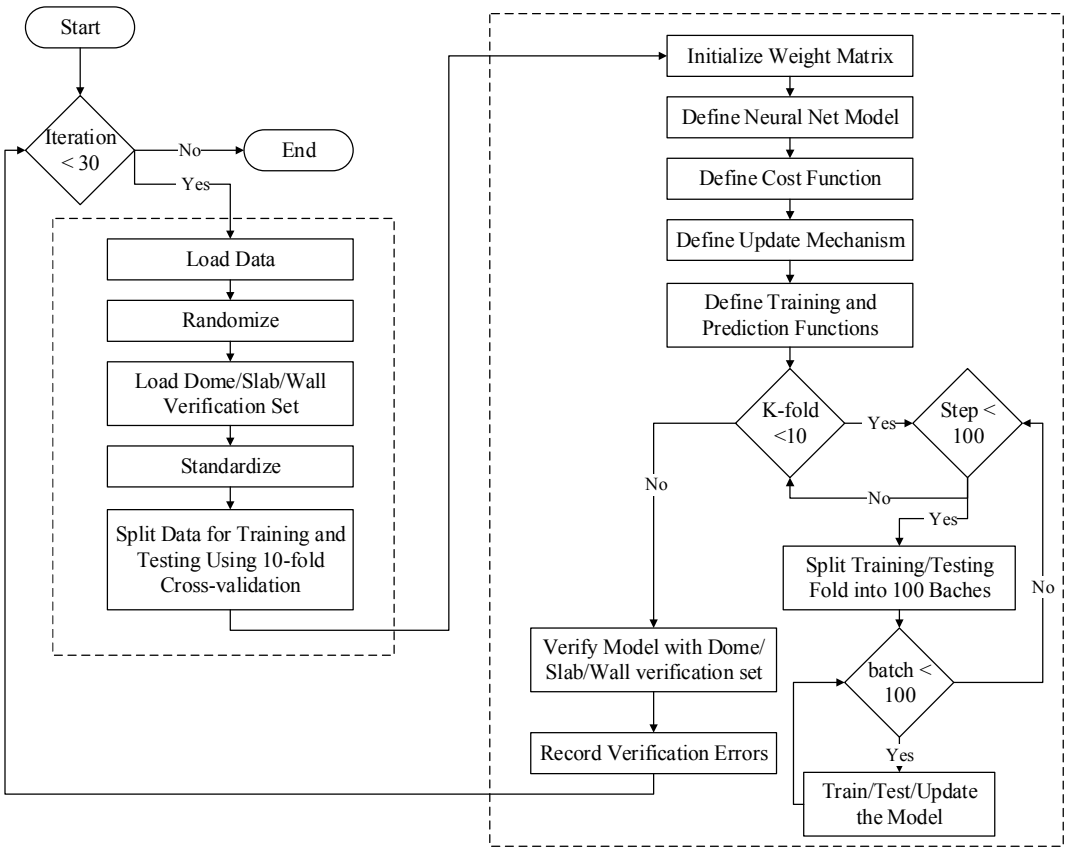


Figure 41: Implementation of Hypothesis 4

Results

The results of experiment show that within each class of structure (Case 1), the slab has the highest percentage of errors in predicting the stress values of unseen structures (4.62%), followed by the dome with 4.5%, and the wall with 1.98%. However, using the data pool (Case 2), the error of the dome and the wall went down by almost 50%, and the slab exhibited the lowest percentage of errors with 0.37% (Figure 42).

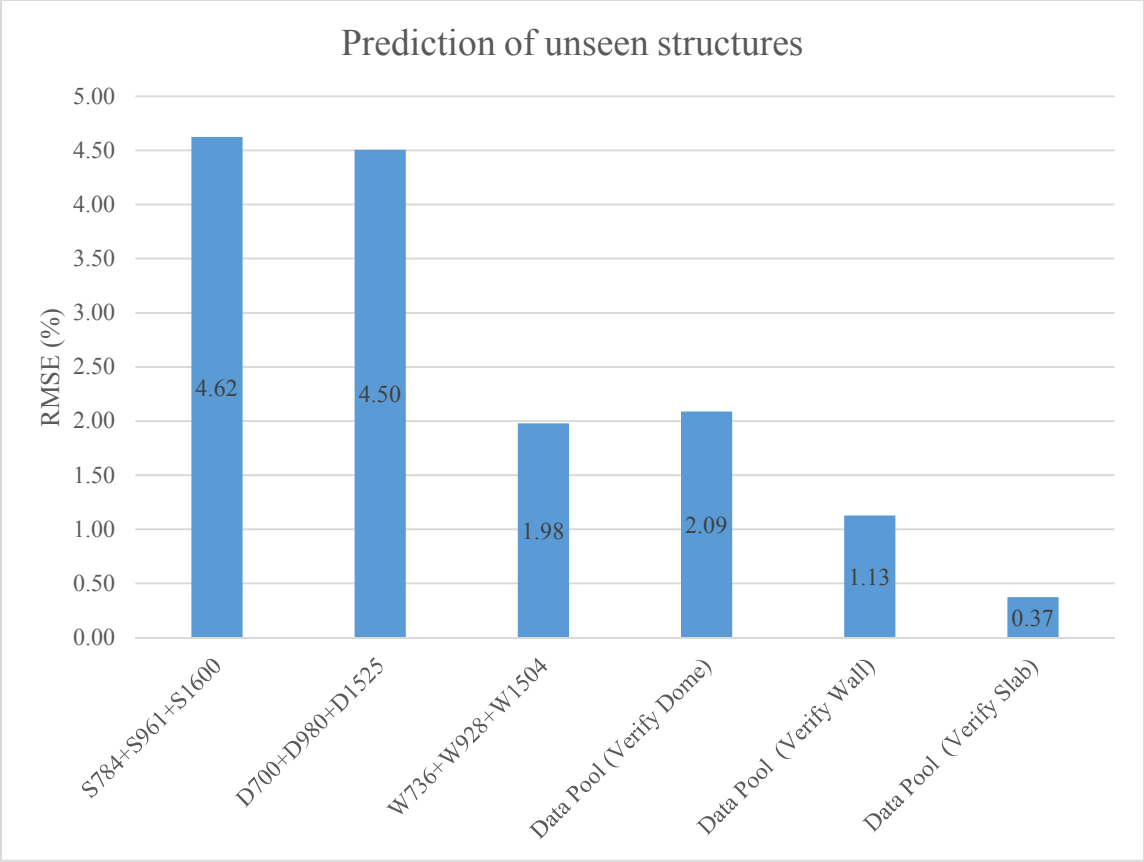


Figure 42: Prediction of the performance of unseen structures

Discussion

The results of this hypothesis underscored the importance of a data pool in predicting the performance of structures that are not included in the training of neural net models. As we combined data from various structures, we were able to more accurately predict the results of finite element analysis for unseen structures.

This study does not measure the relationship between the accuracy of the metamodels and the geometrical properties of the structures. For instance, the metamodel created in Hypothesis 4 is able to predict the results of a 250-member dome. By decreasing the number of dome members, the prediction accuracy of the metamodel may decrease. The metamodels may not work if the number of the structural members is

below a threshold. Future studies can investigate the relationship between the accuracy of the metamodels and the number of structural members.

CHAPTER 5

APPLICATION OF GENERALIZABLE METAMODELS IN BUILDING CONSTRUCTION

This chapter demonstrates the use of metamodels at the early stage of a construction project when the key participants of a project need to make informed decisions about the structural layout of their project. This chapter first describes the current design process of the project and then presents the major findings of applying metamodels in the project.

The Project

The construction project of this case study, located in downtown Los Angeles, California (Figure 43), is at the early stage of the design. The owner of the building, the Museum Tower, has a limited budget and would like to evaluate various design alternatives to make informed decisions about the architectural and structural layout of the new building. Perkins+Will, John A. Martin Associates, Holmes fire & Safety, and Hathaway Dinwiddie are working on the architectural, structural, fire protection, and cost estimation of the project, respectively.

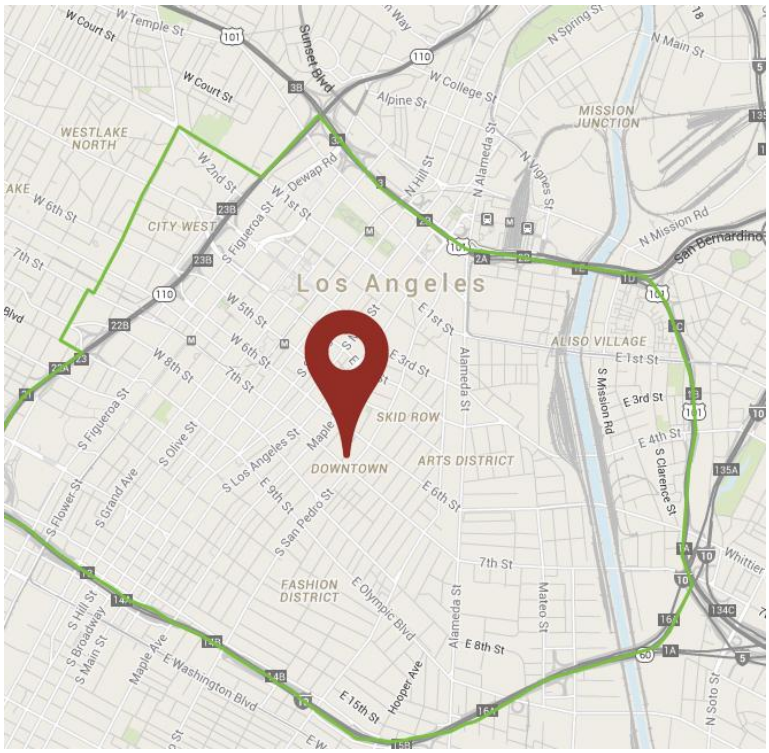


Figure 43: Location of the project in Los Angeles, California

Using the existing reinforced concrete as a benchmark, they are investigating various cost-effective structural and architectural design alternatives. Initially, they were going to study nine designs: three timber and three fixed- and pinned-joint steel structures. Because of the time required to evaluate one design, they could only complete the evaluation of one design.

Architectural Design

The building has 20 identical floors (Figure 44) with a length of 40m (131.23 ft.) and a width of 21m (68.9 ft.). With a typical floor-to-floor height of 3m (9.84 ft.), the overall height of the building is 60m (196.85 ft.).

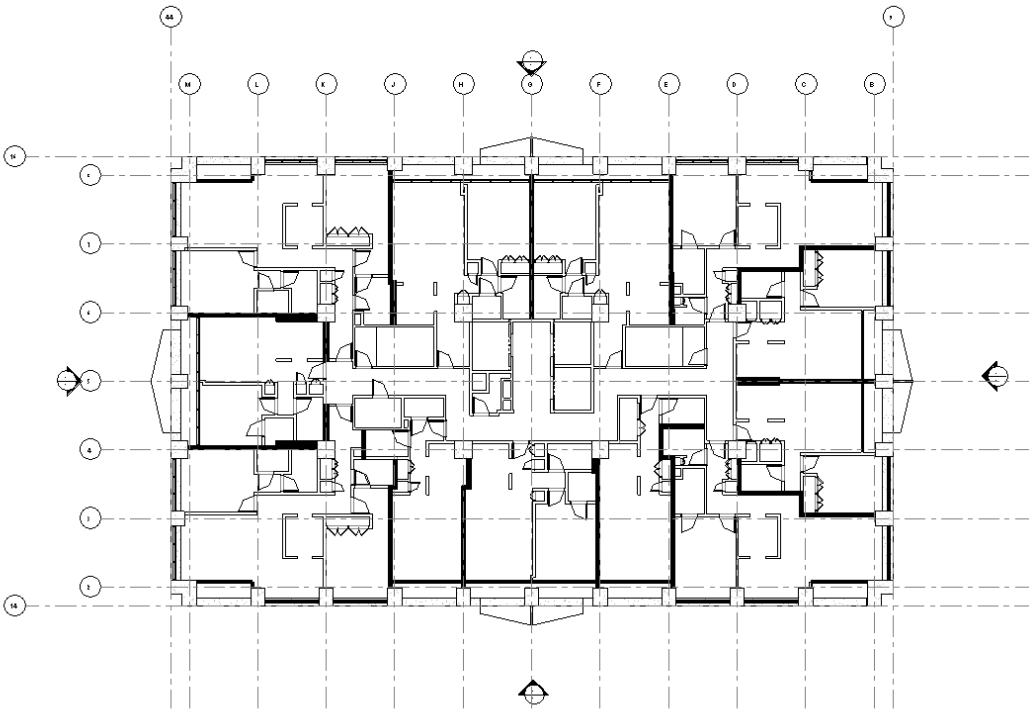


Figure 44: Floor plan of the tower

Figure 45: Perspective view of the tower

Current Design Process

In the current design process, the project planning team, including the Museum Tower and Perkins+Will, designs several architectural layouts that may satisfy the limits of the owner's budget. After creating building forms at the end of the schematic design phase, the design team sends their selected design to structural engineers for calculating the size and the shape of the structure. Engineers create an analytical model to compute the reaction of the structure to loading conditions, compare the results with building codes, and either accept the design or modify the analytic model to obtain acceptable results. At the end of the structural design phase, the design and planning team receives the structural model from the engineers and sends both architectural and structural models to a team that will estimate the project cost. If the cost of the design is higher than the owner's budget, the design team will adjust the design and repeat the process (Figure 46).

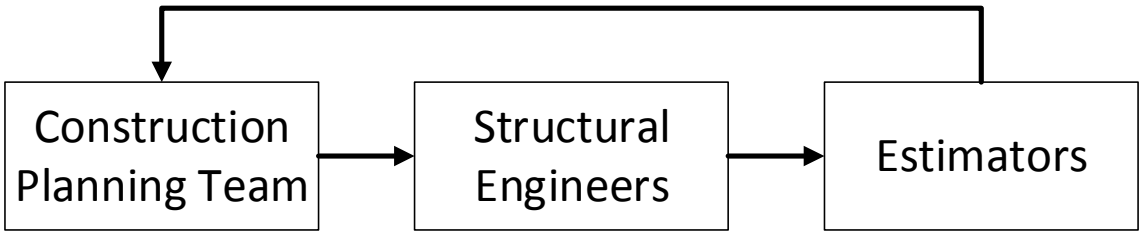


Figure 46: Current design process

The current design process has various limitations. First, the construction planning team can evaluate a limited number of design alternatives because of the time it takes for structural design, analysis, and cost estimation. Currently, the structural engineering and estimating processes take four months (two months each). Second, the current process is repetitive and inefficient especially when the cost of the project is

higher than the budget of the owner. In this case, architects should change the design and send it back to structural engineers and estimators for evaluation. Finally, most of the decisions made at the early stage of the design are based on experience, not the actual construction cost data. To keep the project cost within the owner's budget, instead of proposing a design, waiting for a long time to obtain the cost, and then realizing that the cost of the project is over budget, the construction team needs to include construction costs early on.

Objective of the Case Study

The objective of this case study is to examine the use of metamodels on the evaluation of three structural design alternatives at the early stage of the construction project. By comparing the time of evaluation and the value of the time is for the owner, this study demonstrates the advantages of using metamodels at the early stage of the design for cost-benefit analysis and decision making.

Methodology

To determine which structural layout is the most cost-effective, we use optimization in two cases. In the first case, we evaluate the objective function using a finite element analysis package. In the second case, we evaluate the objective function using metamodels (Figure 44).

Optimization Formulation

Geometry

This case study examines three moment-resistant structural layouts—9x4 (nine blocks by four blocks), 10x5, and 11x6—using one of the best commercially-available

parametric modeling software programs, Autodesk Dynamo. This program has an optimization package called “Optimo” and a structural package called “Structural Analysis for Dynamo,” which sends the parametric model of the structure to “Robot Structural Analysis” software and receives the results of a finite element analysis.

One of the limitations of this workflow is the number of structural members in the analysis: As the number of structural members increases, the possibility of receiving an error in the optimization loop dramatically increases. Because of this limitation, we reduce the number of floors from 20 to ten. This change, in the 9x4 layout, reduces the number of members from 2,700 to 1,350. Figure 47 illustrates the plan and the elevation view of the 9x4 layout, and Figure 48 shows the parametric model generated in Dynamo and Robot Structural Analysis.

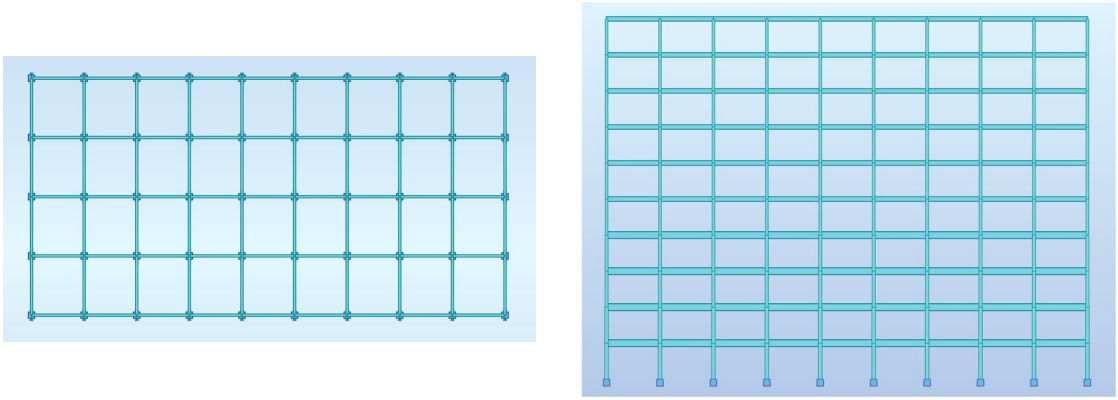


Figure 47: Plan (left) and elevation (right) view of the 9x4 layout

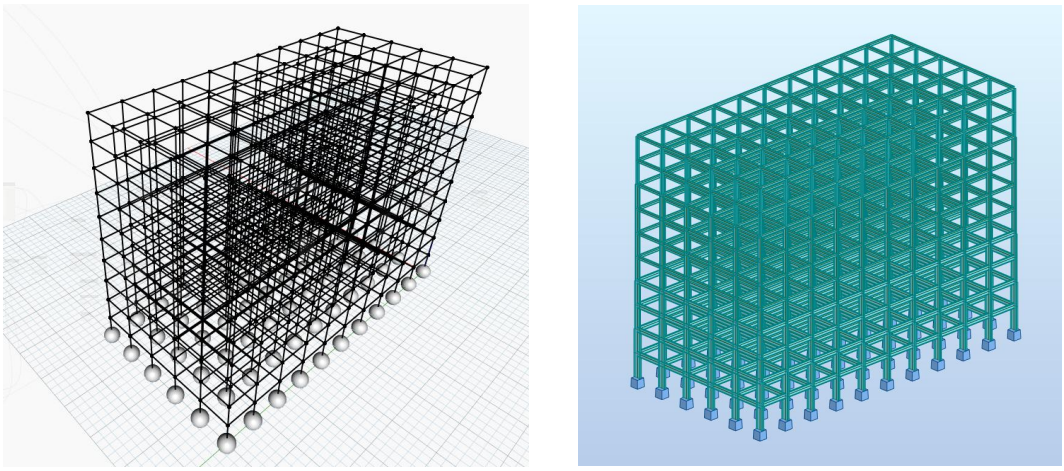


Figure 48: Parametric model in Dynamo (left) and the analytical model in Robot Structural Analysis (right)

Design and Analysis of the Current Apartment Building

The actual residential building of the study is designed and analyzed for wall and floor acoustics, thermal and structural performance, and moisture, UV, and fire protection. In the structural design and analysis, which is the focus of this case study, the frame of the building is designed to comply with allowable stress design, accounting for the stress, deflection, and buckling of structural members. According to ASCE 7-10, loads applied to the structure are dead, live, wind, and seismic loads in various combinations.

Simplification of the Structural Design

The metamodels of this study account for only one load combination. Therefore, a 1.2 dead load + 1.6 live load + 1.0 wind load (applied to the long direction) is selected for the structural analysis. The dead, live, and wind load of the project are 65, 40, and 50 PSF, respectively. Unlike the structural design of the main building, this study does not

design the building for the deflection and buckling of the members. However, we check the compliance of the members with allowable stress design.

Cost of Steel Sections

The cost estimating team of this project includes the following costs for estimating the construction cost of steel sections at the conceptual design stage of the project:

- Cost of the order (e.g., supply, fabrication, detailing)
- Cost of shipping and handling
- Cost of fire protection
- Cost of construction (e.g., labor, erection)

In this case study, with the aid of the one of the structural engineers of the project, we select 16 steel sections and send them to the estimating team for calculating the cost of the members per linear foot. The following table shows the cost of the steel sections.

Table 15: Cost of the steel sections (USD)

Sections	Order	Shipping and Handling	Fire Protection	Construction	Total Cost (Linear feet)
W 10x12	10.91	4.36	9.68	8.18	33.14
W 8x13	11.82	4.73	8.39	8.86	33.80
W 8x18	16.36	6.55	9.37	12.27	44.55
W 12x22	20.00	8.00	11.44	15.00	54.44
W 16x26	23.64	9.45	14.83	17.73	65.65
W 14x30	27.27	10.91	14.40	20.45	73.04

Table 15 (continued)

W 16x40	36.36	14.55	16.10	27.27	94.29
W 21x44	40.00	16.00	19.04	30.00	105.04
W 18x50	45.45	18.18	17.84	34.09	115.57
W 10x54	49.09	19.64	14.08	36.82	119.63
W 12x65	59.09	23.64	16.88	44.32	143.93
W 14x74	67.27	26.91	18.24	50.45	162.88
W 24x76	69.09	27.64	23.04	51.82	171.58
W 16x77	70.00	28.00	18.76	52.50	169.26
W 30x90	81.82	32.73	27.93	61.36	203.84
W 36x150	136.36	54.55	33.47	102.27	326.66

Given the same weight, the cost of lighter sections is higher than that of heavier sections because of the cost of fabrication and detailing (Figure 49).

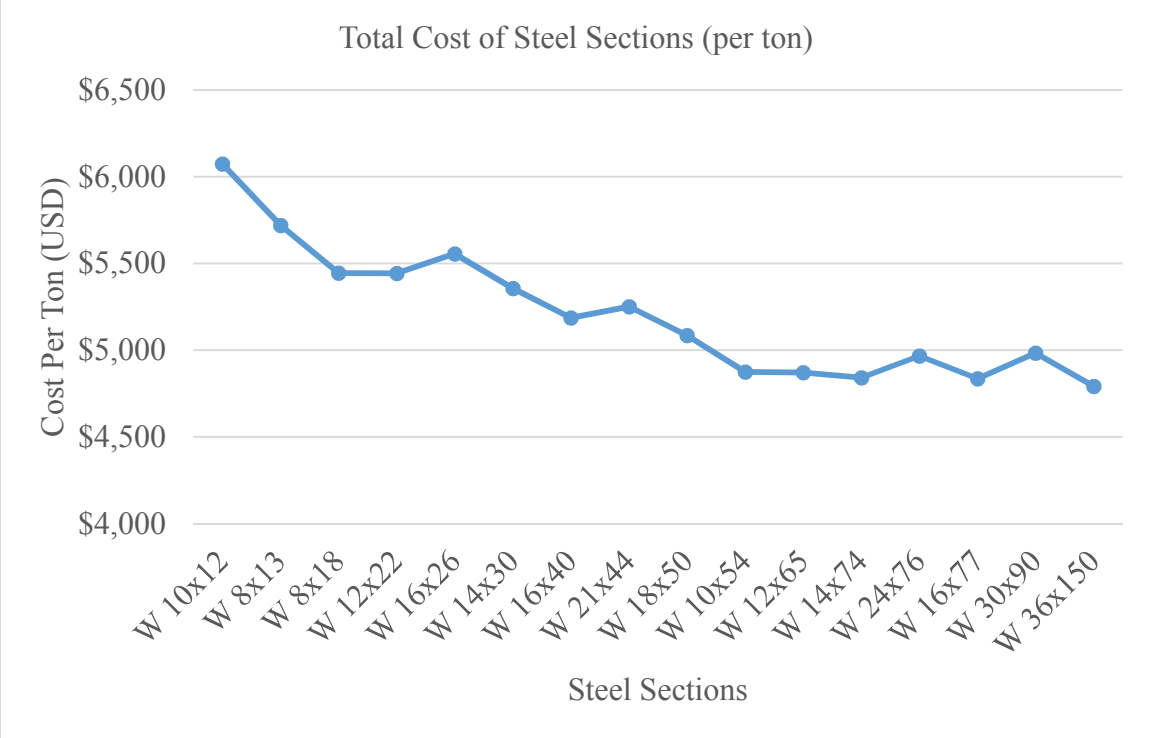


Figure 49: Total cost of steel sections per ton

Mechanical Properties of Steel

The mechanical properties of the steel (A992-Grade 50) used in the structural analysis of this study is as follows:

- Density: 0.2836 lb/in3 (7850 kg/m3)
- Yield strength: 50 ksi (345 MPa)
- Tensile strength: 65 ksi (450 MPa)
- Elongation: 18% in 2"

The maximum allowable stress in the analysis is set to 200 MPa (29 ksi), yielding the safety factor of 2.24.

Optimization Parameters

This section summarizes the optimization parameters of this case study. These parameters are implemented using a genetic algorithm package in Dynamo called Optimo (Figure 50).

Objective: Minimizing construction costs

Variables: Each of the two levels has one section for a beam and one section for a column. Therefore, the entire structure has ten design variables. The number of variables ranges from zero to 15 because we have 16 structural sections in total.

Constraints: The stress of each structural member should be less than the allowable stress.

Fitness Function: The fitness function of this optimization problem is the total cost of structural members. A scaler value as a penalty (i.e., 1,000) is multiplied by the cost of a member when the stress value of the member is higher than the allowable value of stress (i.e., 29 ksi). The implementation of the fitness function is presented in Figure 51.

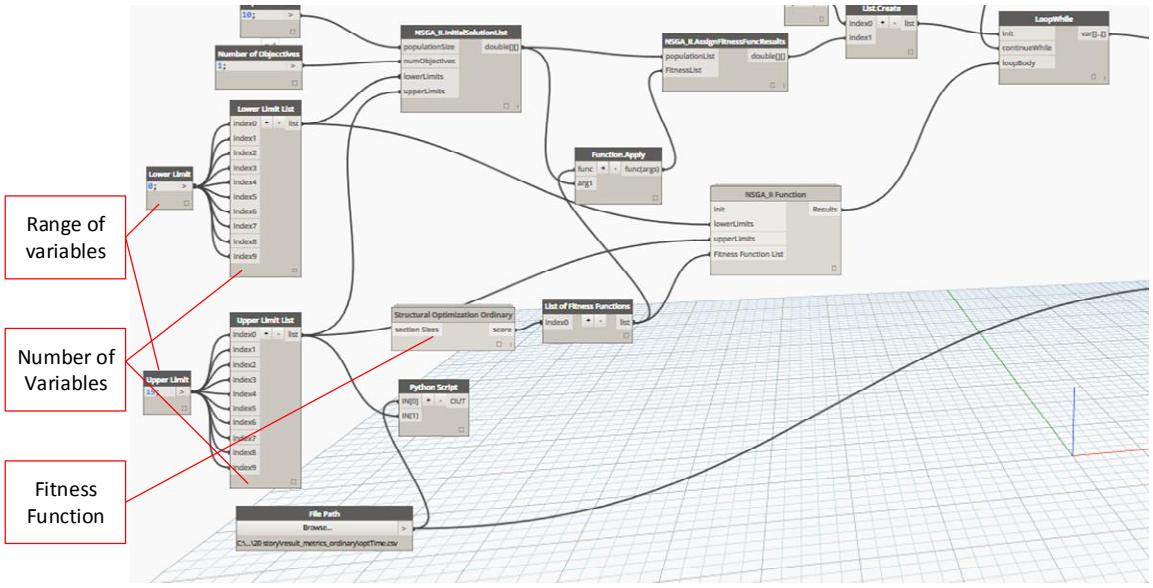


Figure 50: Optimization setup in Dynamo

```
for i in range(len(stress[0])):
    sectionIndex= sectionNo(i)

    secCostIndex= (int) (section[sectionIndex])
    secIndexFeature.append(secCostIndex)
    cost= secCost[secCostIndex]*barLength[i]*3.28084

    if (abs(stress[0][i]/sMax) >1.0):
        cost=cost*1000.0
        noOfPenaltyApplied+=1
    penalizedRatio.append(cost)
```

Figure 51: Implementation of the fitness score

Optimization with Finite Element Analysis

In this case, the objective function of each iteration of the optimization is evaluated by finite element analysis software in the following steps (Figure 52). First, using scripts written in Iron Python language, the geometry of the structure is created. Second, the beams and columns of each two levels are grouped into one category. Third,

selected sections from the genetic algorithm are assigned to these categories. Fourth, using the Structural Analysis for Dynamo package, these geometries and sections are sent to the Robot Structural Analysis. After completion of the analysis, the results are sent back to Dynamo. Finally, using these results, the fitness score of the structure is calculated.

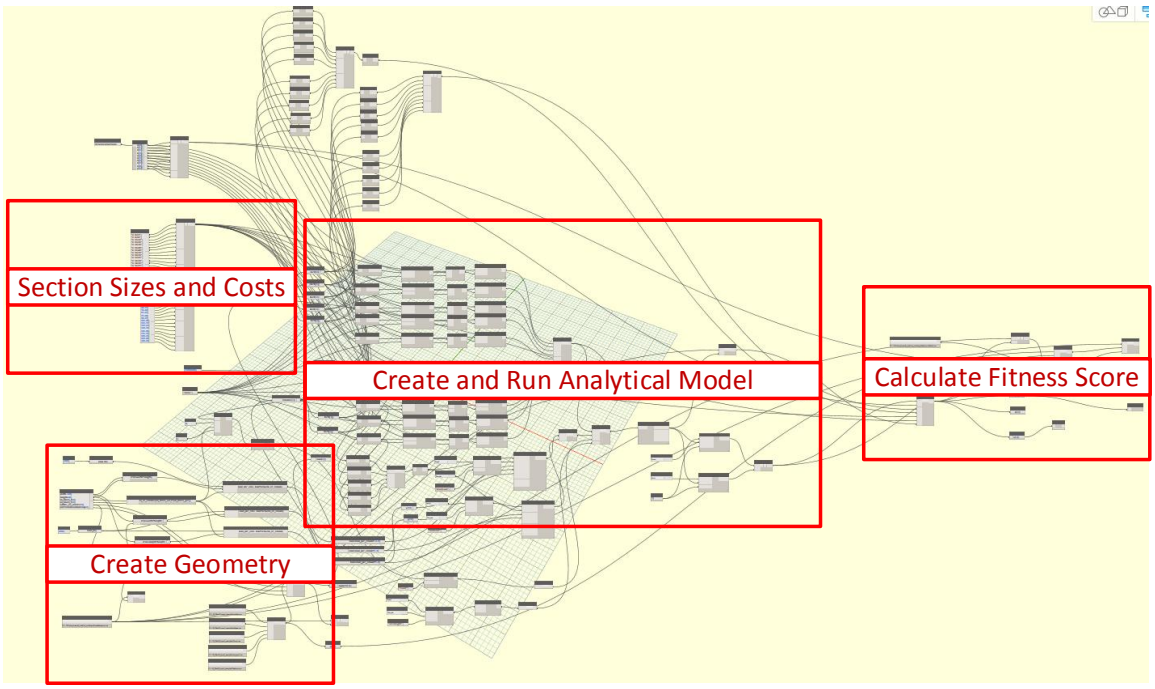


Figure 52: Dynamo graph of the fitness function (with FEA)

Optimization with Metamodels

Optimization with metamodels is similar to optimization with finite element analysis with one major difference: the geometry of the structure is not solved by the finite element analysis software. Instead, the analytical model is solved by metamodels (Figure 53).

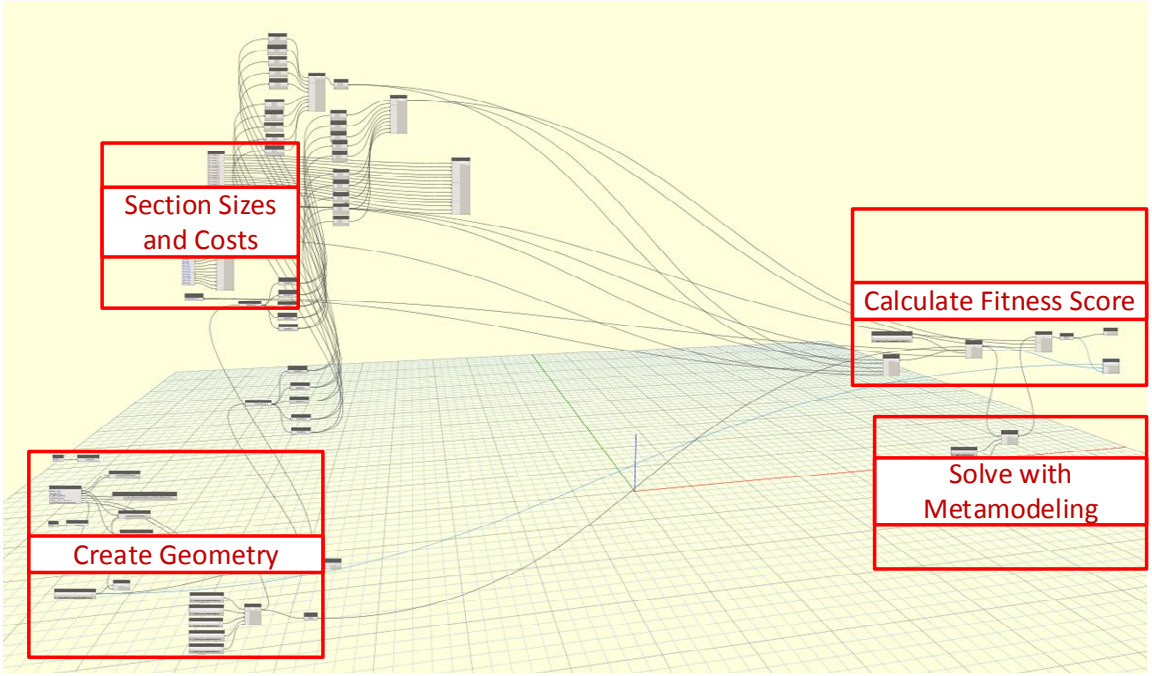


Figure 53: Dynamo graph of the fitness function (with a metamodel)

Results

This section presents the results of a comparison between the time and cost of optimization with analysis software and those of the metamodels. In both cases, a laptop computer, HP 15-j059nr with Intel i7 2.4GH, 8GB RAM and a 2GB Nvidia graphics card, was used. Operations on the CPU were all single-threaded. In all three structural layouts, optimization time with FEA is more than that with the metamodels (Figure 54). The optimization of the 11x6 and 9x4 layout with FEA took almost 24 and 18 hours, respectively. By contrast, the optimization of these layouts with metamodels took slightly less than eight and seven hours.

In both cases, the optimization time includes the time of creating geometry, calculating fitness score, and implementing the genetic algorithm (e.g., selection, mutation, cross-over, and reproduction). However, optimization with the finite element package requires time for creating the members of the structure (bars), sending them to the solver, waiting for the solver to create, test, and solve the model, and receiving the

results from the solver. By contrast, optimization with metamodels consumes time only for converting the model into feature vectors and solving the structure via matrix operations.

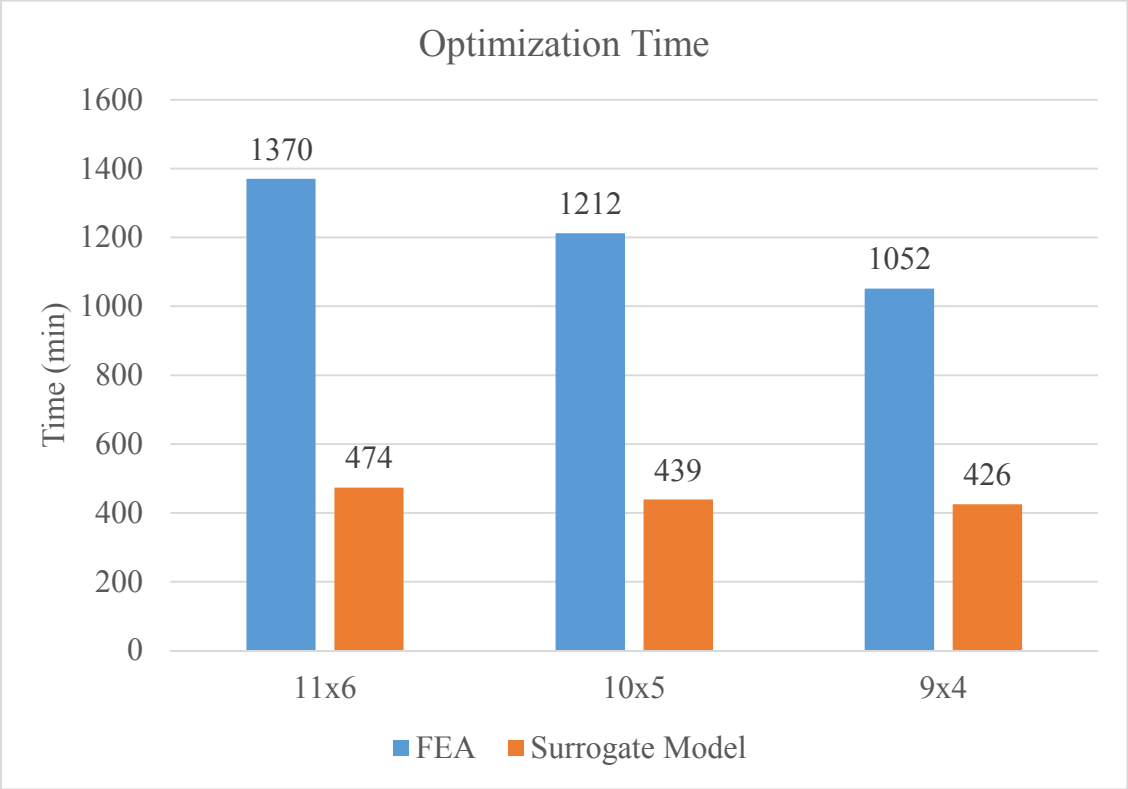


Figure 54: Optimization time

The fitness score of the two optimization techniques are calculated and presented in Figure 55. Given the results of optimization with FEA as the ground truth, the accuracy of the results of optimization with metamodels range from 66% (for 9x4) to 75% (for 11x6). It should be noted that the aim of this case study is not to determine the accuracy of the metamodels, for such a determination would require one to analyze many more cases and samples to reach a statistically significant conclusion.

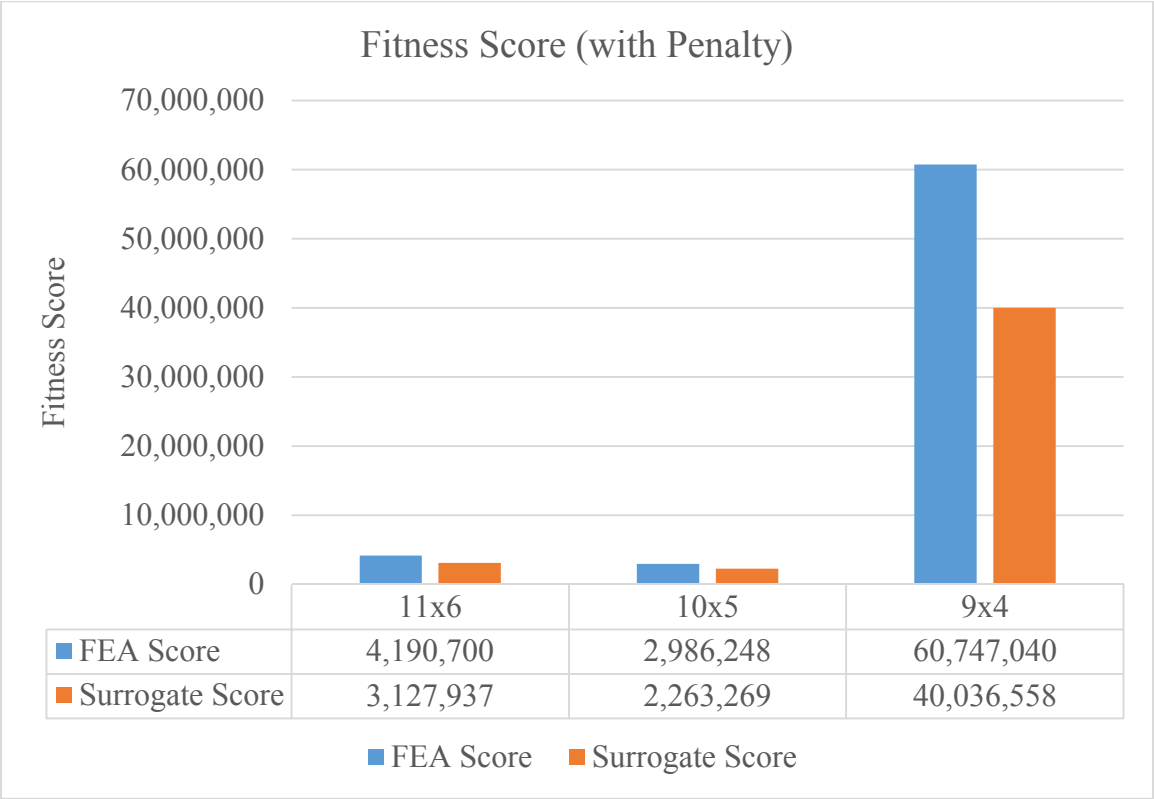


Figure 55: Fitness score

The construction cost of each layout without the penalty weight is presented in Figure 56. The results show that given the 16 available sections, the 9x4 layout has no feasible answer. In addition, they show that because the 10x5 layout costs the least, it is the most cost-effective solution when we use both optimization methods.)

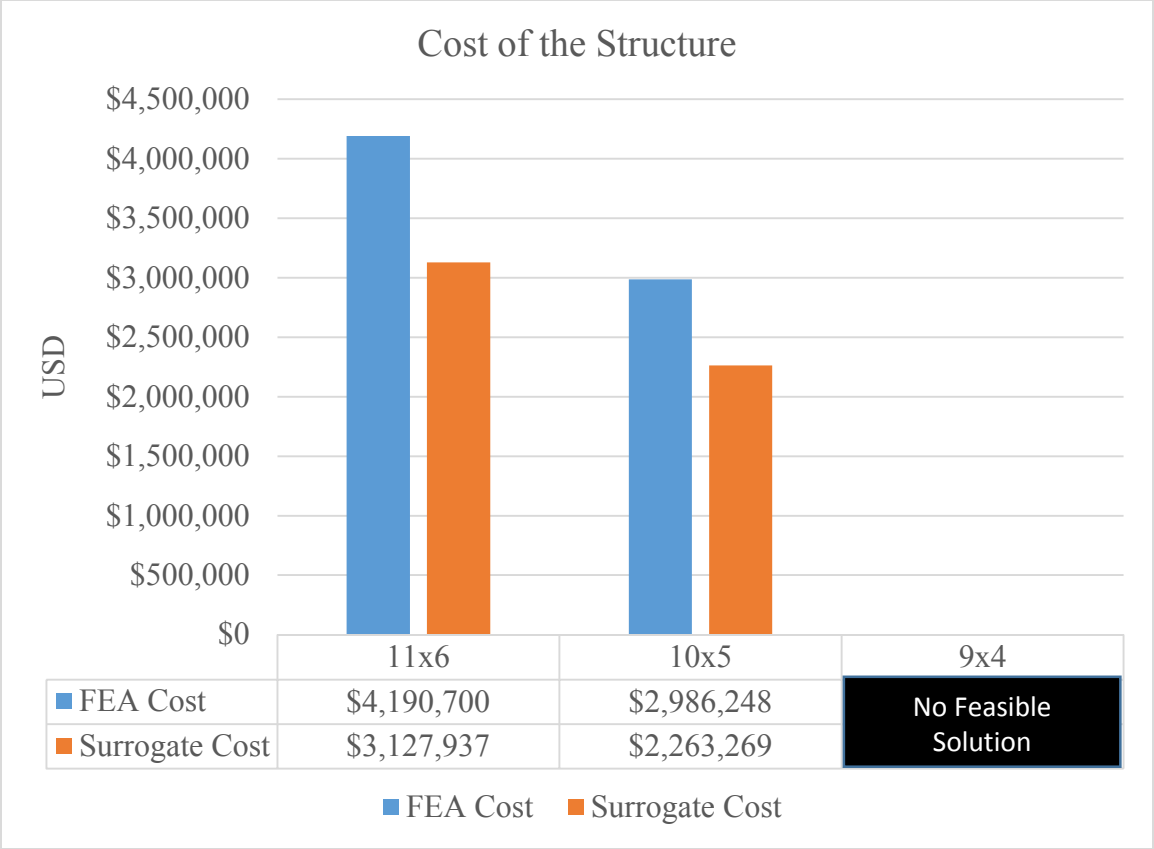


Figure 56: Total cost of the layouts

Quantitative Assessment

A quantitative assessment shows the difference between the time of decision making in the current process, optimization using FEA, and optimization using metamodels (Table 16). In the current process, the evaluation of one structural layout takes four months because of the high level of detail. Since we simplified the structural analysis of the project, we reduce the time of the current process by two months to make a fair assessment. In addition, we increase the time of optimization using both techniques because it is an iterative process that requires various iterations to find a satisfactory answer. We also assume that the time required to assess more than one layout is half that of assessing the first layout. For example, if the time for assessing an 11x6 layout is one month, the time for assessing a 10x5 would be two weeks. This reduction in time is the

result of engineers' and estimators' becoming familiar with the design, which allows them to design and estimate the cost of a project in a shorter time.

Table 16: Decision-making time

Time	Current Process	Optimization with FEA	Optimization with Metamodels
One Layout	Two months	seven days	One day
Three Layouts	Four months	14 days	Two days
Nine Layouts	Ten months	35 days	Five days

The opportunity cost of the time spent on the evaluation of the various layouts in this project is $500,000 per month (Table 18). In addition, the owner pays almost $100,000 per month to the team of consultants. Therefore, the opportunity cost for the owner is $20,000 per day, money that the owner is losing each day the project is delayed.

Table 17: Opportunity cost for the owner (USD)

Cost	Current Process	Optimization with FEA	Optimization with Metamodels
One Layout	$1,200,000	$140,000	$20,000
Three Layouts	$2,400,000	$280,000	$40,000
Nine Layouts	$5,300,000	$700,000	$100,000

Qualitative Assessment

In addition to the quantitative assessment, the results of the study are presented to the construction team. The following section presents the results of this qualitative assessment.

Optimization and decision making with metamodels is an extra step that requires all of the project participants to gather together at the early stage of the design (Figure 57). They may need a week to make decisions about various design alternatives. Then, selected alternatives are sent to engineers and cost estimators for confirmation. The current design process offers no assurance that the design meets the owner's budget. However, using metamodels contributes to ensuring that the design cost will remain within the budget.

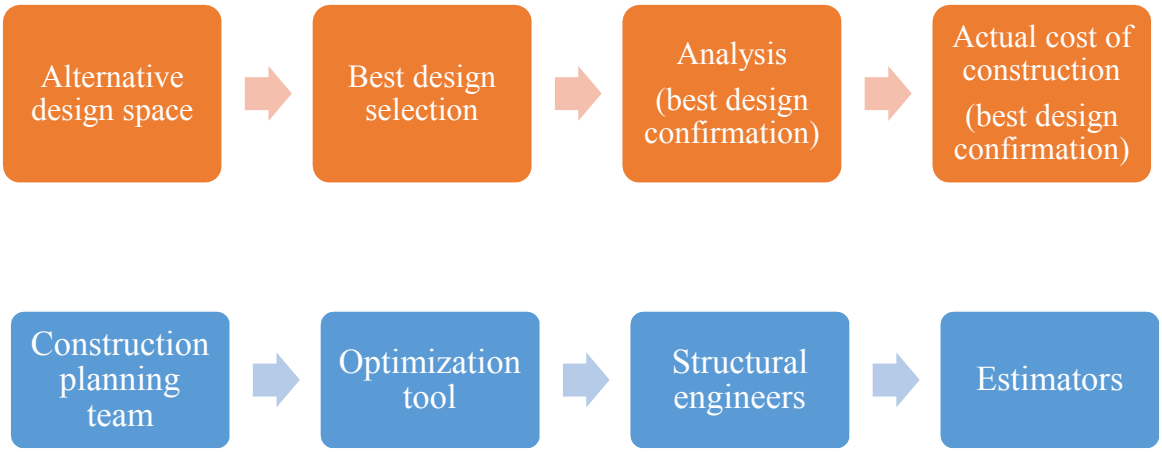


Figure 57: Change in the current process

Limitations

One of the major limitations of this study was that it reduced the number of stories from 20 to ten because of the limitations of the analysis packages. In addition, the structural design and analysis involved the following simplifications:

- Only dead load, live load, and wind are applied to the structure (no earthquake load).
- Wind load is only a pressure, normal to the surface and does not include suction forces.

- Only one load case (1.2 DL + 1.6 LL + 1.0 W) is applied to the structure.
- The structure is analyzed in the elastic mode and the analysis does not reflect the non-linear behavior of the structure.
- The analysis does not consider the controlling load case of the structure for wind or earthquakes in the short and long directions. The only lateral load (i.e., wind load) is applied to the long direction of the structure.
- The design is only based on the stress values and does not include any current code requirements.
- Connections are assumed to be ideal and not governing in the design of sections.
- Splices are at the level of the floor, not midway between the members as is typical.
- The analysis does not account for the buckling of members and overall stability of the structure.
- The serviceability conditions (e.g., global or local deflection, deformation, and vibration) are not checked.

Discussion and Conclusion

The time it takes for optimization depends on the computing power of the machine that runs the problem. Imagine that we have infinite computing power and are able to run both optimization techniques with thousands of computers. On the one hand, if an optimization process uses FEA solvers to find the solution, for every minimal change in the problem definition, we need to run the optimization from the beginning because this mechanism does not record the history of what has happened in the past. On the other hand, if the optimization process uses metamodels, machines can learn with

significantly improved predictive performance of structures by continuously receiving and aggregating data from various structures.

Using the power of cloud computing and artificial intelligence, users can obtain real-time feedback on their designs. For instance, as architects design a building, they can receive real-time recommendations related to the structural layout of their design. Therefore, they are aware of the cost tradeoffs of their designs, so they can make better decisions at a very early stage of construction projects. Indeed, the goal of this system is not to replace structural engineering and cost estimating efforts, but instead, to show the value of each design alternative so that architects can make informed decisions.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

Using the artificial neural network and back propagation technique, we demonstrated that the results of the analysis of structures with various shapes and topologies could be combined and re-used for inferring the performance of structures within the same geometry class. That is, unlike exclusive models proposed in various studies in the literature, these generalizable neural network models were not limited to predicting the results of only one structure. In fact, they were able to receive data from various structures and predict the results of structures within the same geometry class.

Models developed with the novel features introduced in this study incorporate the analysis results of various structures. Therefore, receiving more data and becoming smarter as more data from users become available. Researchers and designers can use the method of developing the metamodels introduced in this study to create predictive models for the approximation of finite element analysis in structural evaluation of trusses and space frames.

The case study of this research demonstrated the use of generalizable metamodels at the early stage of the construction project. The results illustrated that the optimization of structures using the generalizable metamodels takes less time compared to using the FEA solver: 35%, 36%, and 40% with the accuracy of 75%, 76%, and 66% for the 11x6, 10x5, and 9x4 layout respectively. Using generalizable metamodels, the construction planning team can quickly evaluate various structural design alternatives at the early stage of construction projects.

Contributions of the Research

The contributions of this study to the body of knowledge are as follows:

- Feature descriptors that enable the aggregation of structural analysis data from various construction projects
- A methodology that creates reusable metamodels that enables the more practical optimization of structures for use in the construction industry
- The optimization of structures that reduces the time and cost of construction projects
- A methodology that fosters more efficient planning and more rapid decision-making by architects, contractors, and engineers at the early stage of (or during) construction projects

Limitations of the Study

This study is limited in several ways. First, the metamodels introduced in this study were created and tested using three classes of regular-shape geometries. In addition, forces applied to the structures had a fixed magnitude and direction in order to prevent the curse of dimensionality. Another limitation is that only one load case was applied to the structures; lateral loads (e.g., wind, earthquake tremors), however, were not applied to the structures for testing the four hypotheses. Finally, the structural analysis did not account for either buckling or displacement of the structure.

Recommendations for Future Research

This study was the first step toward developing generalizable metamodels able to predict the performance of various structures. In future research, this study could expand

in various ways. For one, because the metamodels of this study were built and tested using three types of structures, studying a variety of structures could lead to a more complete understanding of the problem. In addition, since the geometries of this research had a regular shape, the creation and testing of metamodels for structures with irregular geometry could extent the generalizability of this models. An additional direction of research would be to introduce new features that describe the geometry and then examine the interrelationship among the features, which may improve the performance of the metamodels. Another recommendation is to test the metamodels in various optimization problems either as a surrogate of finite element analysis or an assistant to the solver.

Implications of the Current Research

Traditionally, the process of structural optimization used to be a time consuming manual task. Architects have more flexibility to change their designs during the early stage of a building project, when they make the most critical decisions about the geometry, mass, and forms of buildings (Wang, Shen et al. 2002). However, engineers commonly analyze the structure of buildings after the schematic design stage, when the architects have less flexibility and changes to their designs cost more. To alleviate this problem, researchers at HOK proposed a new strategy called "shifting the effort," a concept that refers to focusing on the redistribution of efforts at the earliest stage of projects to improve the outcome of design processes.

Since most design variables are not well-defined at the conceptual stage of projects, optimizations help designers gain a more thorough understanding of their design space and a wider selection of design variables. For example, in a construction project, both wide flange (WF) and H sections could be available. The optimization of beams could find that WF sections are the best solution based on the boundary conditions of the

structure. Because of the vague definition of "design variables," the accuracy of the results of optimization is not important. Therefore, designers can use metamodels to approximate the results of time-consuming structural optimization problems. This study approximated the results of optimization by demonstrating a method of creating generalizable metamodels that are re-usable in size optimization problems. In addition to the optimization of structures, the proposed metamodels have the following benefits for the construction industry.

Construction Estimation of Structural Members

Since detailed designs do not exist at the conceptual design stage, the cost estimation of projects is more likely to be based on rules of thumbs and expert judgment, both of which are time-consuming and subject to error (Murat Günaydın and Zeynep Doğan 2004, Aram 2014). In this case, estimators predict the cost of projects based on their records from previous projects, not on the real size and shape of building elements, so the predicted cost will differ from the actual cost. To minimize the difference between estimators' prediction of the sizes of building members and their actual size and maximize the accuracy of the predictions so that they are close-to-actual cost, estimators can use the metamodels proposed in this study to estimate the cost of projects based on the size and shape of building elements.

Bidding and Production Planning

Another benefit of metamodels is that they assist contractors with preparing bid documents and planning for construction projects. For instance, by calculating the size of building members with metamodels in the early stage of projects, they can more closely estimate the cost of the members by including the costs of machinery and for the

handling and transportation of members (e.g., number of trucks and type and capacity of cranes) and thus prepare a stronger bid. In addition, they can make informed decisions when choosing fabricators based on the size of structural members.

Process Improvement

The structural design process currently requires multiple design exchanges between architects and structural engineers. After creating building forms at the end of the schematic design phase, architects send the design to structural engineers, who compute the size and shape of structures. Then the engineers create an analytical model from which they calculate the reaction of the structure to loading conditions, compare the results with building codes, and either accept the design or modify the analytic model to obtain better results. Sometimes the size and the location of structural frames resulting from the structural analysis interfere with the forms that architects, who estimate the size of beams and columns, have designed. By employing generalizable metamodels for the approximation of building elements at the early stage of design, architects can more easily and accurately estimate the size of structural elements, which reduces the number of design exchanges between the structural engineers and architects.

Communication Enhancement

Using metamodels in construction projects can enhance communication among estimators, structural engineers, and architects. For instance, through improved communication, estimators and architects can follow the same analysis logic as structural engineers so that the structural design more closely matches the detailed design. In this case, if possible, structural engineers can advise other parties and help them set up the

structural optimization problem. Using metamodels would therefore shift the availability of structural design to the very early stage of construction projects.

Structural Design and Optimization

Optimization is an iterative process that begins when designers and engineers first define their optimization problems at the early stages of projects, when they have only a vague idea of the problem. After running the optimization problem and waiting for several hours, days, or even weeks, they attempt to explore the design space and understand the relationship between their input and output variables. Then engineers refine and rerun their problem, a process that repeats until the designers are satisfied with their results.

In the current practice of construction, structural engineers commonly have a limited time to analyze structures and complete the detailed project design. Therefore, the time-consuming iterative process of the optimization of structures is not feasible. A review of the literature that compared the results of manual and computerized engineering optimization found that manual optimization takes 30 fewer man hours than computerized optimization because it does not require setting up and running the optimization problem. However, the quality of the computerized process was much better, saving $1.2 million dollars (Flager, Adya et al. 2014). In addition, the iterative process of optimization can significantly delay the design and the construction of projects. For instance, if an optimization problem with four problem definition refinements takes seven days to run, it will take 28 days to complete. If the metamodels proposed in this study are used, the optimization of buildings becomes more timely and economically viable because the metamodels are generalizable; that is, they can be re-used in various projects.

Optimization of 3D-Printed Buildings

The future method of constructing buildings using advanced manufacturing will become more efficient and faster than the conventional method of construction. In this case, even though the optimization of such structures saved millions of dollars, the use of 3D-printers remains limited because of their inability to optimize these structures in a timely manner. For example, cities plagued by earthquakes or hurricanes require buildings with customized shapes and structures. The optimization of these structures in a timely manner would significantly reduce both the time and the cost it takes to complete projects.

For the timely optimization of 3D-printed buildings, this study developed various generalizable metamodels designed for receiving, combining, and predicting the analysis results of various structures. By continuously receiving and aggregating data from various structures, such metamodels show strong predictive power if they are used in the cloud, which enables designers and engineers to complete their optimization problems in a timely manner. Therefore, the use of additive manufacturing would be more practical.

Generalizable Metamodels Beyond the Prediction of Structural Analysis

Even though this study demonstrated a method of creating reusable metamodels for predicting the performance of structural members, the use of these models is not limited to the prediction in a finite element analysis. Using the same technique proposed in this study, engineers can train their models based on the actual cost of building members (e.g., shipping, handling, joint type, and fabrication) and predict the most cost-effective configuration of structures based on loading conditions.

In addition to cost, the constructability of the buildings can be modeled by incorporating the ease of fabrication of each building member into the training of metamodels. Various efforts at calculating the constructability of building members can

be used to qualitatively assess the constructability of buildings (Nourbakhsh, Mydin et al. 2012). Using these metamodels for optimization, architects can generate the most construction-friendly configuration of structures at a very early stage of design.

REFERENCES

Allaire, G., F. Jouve and A.-M. Toader (2002). "A level-set method for shape optimization." Comptes Rendus Mathematique **334**(12): 1125-1130.

Allaire, G., F. Jouve and A.-M. Toader (2004). "Structural optimization using sensitivity analysis and a level-set method." Journal of Computational Physics **194**(1): 363-393.

Allison, J. T., A. Khetan and D. Lohan (2013). Managing variable-dimension structural optimization problems using generative algorithms. the Proceedings of the 10th World Congress on Structural and Multidisciplinary Optimization, Orlando, FL.

Aram, S. (2014). A Knowledge-based system framework for semantic enrichment and automated detailed design in the AEC projects. PhD, Georgia Tech.

ASTM (2012). Standard Terminology for Additive Manufacturing Technologies. West Conshohocken, PA, ASTM International. **F2792-12a**.

Baldock, R., K. Shea and D. Eley (2005). Evolving Optimized Braced Steel Frameworks for Tall Buildings Using Modified Pattern Search. Computing in Civil Engineering**:** 1-12.

Bastien, F., P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley and Y. Bengio (2012). "Theano: new features and speed improvements." arXiv preprint arXiv:1211.5590.

Bendsøe, M. P. (1989). "Optimal shape design as a material distribution problem." Structural optimization **1**(4): 193-202.

Bendsøe, M. P. (1995). Optimization of Structural Topology, Shape, and Material. Heidelberg, Springer.

Bendsoe, M. P. and O. Sigmund (2003). Topology optimization: theory, methods and applications, Springer.

Bendsøe, M. P. and O. Sigmund (1999). "Material interpolation schemes in topology optimization." Archive of Applied Mechanics **69**(9-10): 635-654.

Bishop, C. M. (2006). Pattern recognition and machine learning, springer.

Bletzinger, K.-U., M. Firl, J. Linhard and R. Wüchner (2010). "Optimal shapes of mechanically motivated surfaces." Computer Methods in Applied Mechanics and Engineering **199**(5–8): 324-333.

Cagan, J., D. Degentesh and S. Yin (1998). "A simulated annealing-based algorithm using hierarchical models for general three-dimensional component layout." Computer-Aided Design **30**(10): 781-790.

CanalHouse. (2014). "3D printed Canal House." Retrieved October, 2014, from http://3dprintcanalhouse.com/.

Chang, P.-C. and A. Swenson (2014). Building construction. Encyclopedia Britannica. Online.

Cho, Y. S., L. Xia, S. U. Hong, S. B. Kim and J. S. Bae (2007). "Study of Optimized Steel Truss Design Using Neural Network to Resist Lateral Loads." Key Engineering Materials **348**: 405-408.

Coello, C. A. C., D. A. Van Veldhuizen and G. B. Lamont (2002). Evolutionary algorithms for solving multi-objective problems, Springer.

Corbusier, L. (1986). Towards a new Architecture. New York, Dover Publications, Inc.

Ding, Y. (1986). "Shape optimization of structures: a literature survey." Computers & Structures **24**(6): 985-1004.

Dini, E. (2014). "D_Shape." Retrieved 12/24/2014, from http://www.d-shape.com/.

Eschenauer, H. A. and N. Olhoff (2001). "Topology optimization of continuum structures: A review*." Applied Mechanics Reviews **54**(4): 331-390.

FastCodeSign. (2015). "This Robot Can 3-D Print A Steel Bridge In Mid-Air." Retrieved 09/18/2015, 2015, from http://www.fastcodesign.com/3047350/this-robot-can-3-d-print-a-steel-bridge-in-mid-air#16.

Fieldsend, J. E. and S. Singh (2002). A Multi-objective algorithm based upon particle swarm optimization, an efficient data structure and turbulence. In Proceedings of the 2002 UK workshop on computational intelligence, Birmingham, UK.

Flager, F., A. Adya, J. Haymaker and M. Fischer (2014). "A bi-level hierarchical method for shape and member sizing optimization of steel truss structures." Computers & Structures **131**(0): 1-11.

Flager, F., G. Soremekun, A. Adya, K. Shea, J. Haymaker and M. Fischer (2014). "Fully Constrained Design: A general and scalable method for discrete member sizing optimization of steel truss structures." Computers & Structures **140**(0): 55-65.

Forrester, A., A. Sobester and A. Keane (2008). Engineering design via surrogate modelling: a practical guide, John Wiley & Sons.

Forrester, A. I. J. and A. J. Keane (2009). "Recent advances in surrogate-based optimization." Progress in Aerospace Sciences **45**(1–3): 50-79.

Friedman, D. (2010). Historical building construction: design, materials, and technology. New York, WW Norton & Company.

Gane, V. and J. Haymaker (2012). "Design Scenarios: Enabling transparent parametric design spaces." Advanced Engineering Informatics **26**(3): 618-640.

Goldin, M. (2014). "Chinese Company Builds Houses Quickly With 3D Printing." Retrieved October, 2014, from http://mashable.com/2014/04/28/3d-printing-houses-china/.

Gustafson, M. (2010). Consideration of Practical Design Issues in Formulating Structural Optimization for Design Automation. Structures Congress 2010, American Society of Civil Engineers**:** 467-479.

Hajela, P. and L. Berke (1992). "Neural networks in structural analysis and design: An overview." Computing Systems in Engineering **3**(1–4): 525-538.

Hasançebi, O. (2007). "Optimization of truss bridges within a specified design domain using evolution strategies." Engineering Optimization **39**(6): 737-756.

Hasançebi, O., S. Çarbaş, E. Doğan, F. Erdal and M. P. Saka (2009). "Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures." Computers & Structures **87**(5–6): 284-302.

HPC. (2011). "MSC Nastran Performance Benchmark and Profiling." Retrieved 09/02/2015, 2015, from http://www.hpcadvisorycouncil.com/pdf/MSC_Nastran_Analysis_and_Profiling_AMD.pdf.

Imai, K. and L. A. Schmit Jr (1981). "Configuration Optimization of Trusses." Journal of the Structural Division **107**(5): 745-756.

Imam, M. H. (1982). "Three-dimensional shape optimization." International Journal for Numerical Methods in Engineering **18**(5): 661-673.

Jenkins, W. M. (1999). "A neural network for structural re-analysis." Computers & Structures **72**(6): 687-698.

Kamyab Moghadas, R., K. K. Choong and S. Bin Mohd (2012). "Prediction of optimal design and deflection of space structures using neural networks." Mathematical Problems in Engineering **2012**.

Kaveh, A. and H. Servati (2001). "Design of double layer grids using backpropagation neural networks." Computers & Structures **79**(17): 1561-1568.

Khodadadi, A. and P. von Buelow (2014). "Performance Based Exploration Of Generative Design Solutions Using Formex Algebra." International Journal of Architectural Computing **12**(3): 321-338.

Khoshnevis, B. (1999). Contour crafting – state of development. Proceedings of the Solid Freeform Fabrication.

Khoshnevis, B. (2004). "Automated construction by contour crafting—related robotics and information technologies." Automation in Construction **13**(1): 5-19.

Kicinger, R. P., T. Arciszewski and K. A. De Jong (2005). "Emergent Designer: An integrated research and design support tool based on models of complex systems." ITcon **10**: 329-347.

Kirkpatrick, S., C. D. Gelatt and M. P. Vecchi (1983). "Optimization by Simulated Annealing." Science **220**(4598): 671-680.

Kirsch, U. (1993). Structural optimization, Springer.

Kripakaran, P., A. Gupta and J. W. Baugh Jr (2007). "A novel optimization approach for minimum cost design of trusses." Computers & Structures **85**(23–24): 1782-1794.

Leonhardt, F. (2010). Reflections on 60 years of structural development. Structural engineering: History and development. R. Milne, CRC Press**:** 6-7.

Lim, S., R. A. Buswell, T. T. Le, S. A. Austin, A. G. F. Gibb and T. Thorpe (2012). "Developments in construction-scale additive manufacturing processes." Automation in Construction **21**(0): 262-268.

Lim, S., T. Le, J. Webster, R. Buswell, S. Austin, A. Gibb and T. Thorpe (2009). Fabricating construction components using layer manufacturing technology. Global Innovation in Construction Conference 2009 (GICC'09). Loughborough University.

Liu, M., S. Burns and Y. Wen (2006). "Genetic Algorithm Based Construction-Conscious Minimum Weight Design of Seismic Steel Moment-Resisting Frames." Journal of Structural Engineering **132**(1): 50-58.

Mayer, S. (1998). Distributed Parallel Solution of Very Large Systems of Linear Equations in the Finite Element Method, Herbert Utz Verlag.

McCulloch, W. S. and W. Pitts (1943). "A logical calculus of the ideas immanent in nervous activity." The bulletin of mathematical biophysics **5**(4): 115-133.

Miguel, L. F. F., R. H. Lopez and L. F. F. Miguel (2013). "Multimodal size, shape, and topology optimisation of truss structures using the Firefly algorithm." Advances in Engineering Software **56**(0): 23-37.

Mueller, C. T. (2014). Computational exploration of the structural design space. PhD, Massachusetts Institute of Technology.

Murat Günaydın, H. and S. Zeynep Doğan (2004). "A neural network approach for early cost estimation of structural systems of buildings." International Journal of Project Management **22**(7): 595-602.

Mx3D. (2015). "MX3D Bridge." Retrieved 9/12/2015, 2015, from http://mx3d.com/.

Nastran. (2014). "MSC Nastran 2014.0 Test Results." Retrieved 02/09/2015, 2015, from http://web.mscsoftware.com/support/prod_support/nastran/performance/msc20140.cfm.

Nourbakhsh, M., S. H. Mydin, R. M. Zin, S. Zolfagharian, J. Irizarry, M. Zahidi, X. Du, J. Zheng, W. Yan, Y. Li and J. Zhang (2012). "A Conceptual Model to Assess the Buildability of Building Structure at Design Stage in Malaysia." Trends in Civil Engineering, Pts 1-4 **446-449**: 3879-3884.

Pedersen, N. L. and A. K. Nielsen (2003). "Optimization of practical trusses with constraints on eigenfrequencies, displacements, stresses, and buckling." Structural and Multidisciplinary Optimization **25**(5-6): 436-445.

Priest, J. W., C. Smith and P. DuBois (1997). Liquid metal jetting for printing metal parts. Solid Freeform Fabrication Proceedings, University of Texas at Austin, TX.

Queipo, N. V., R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan and P. Kevin Tucker (2005). "Surrogate-based analysis and optimization." Progress in Aerospace Sciences **41**(1): 1-28.

Querin, O. M., G. P. Steven and Y. M. Xie (1998). "Evolutionary structural optimisation (ESO) using a bidirectional algorithm." Engineering Computations **15**(8): 1031-1048.

Ramasamy, J. V. and S. Rajasekaran (1996). "Artificial neural network and genetic algorithm for the design optimizaton of industrial roofs —A comparison." Computers & Structures **58**(4): 747-755.

Salajegheh, E. and S. Gholizadeh (2005). "Optimum design of structures by an improved genetic algorithm using neural networks." Advances in Engineering Software **36**(11–12): 757-767.

Shea, K., R. Aish and M. Gourtovaia (2005). "Towards integrated performance-driven generative design tools." Automation in Construction **14**(2): 253-264.

Shea, K. and J. Cagan (1999). "The design of novel roof trusses with shape annealing: assessing the ability of a computational method in aiding structural designers with varying design intent." Design Studies **20**(1): 3-23.

Shea, K., J. Cagan and S. J. Fenves (1997). "A Shape Annealing Approach to Optimal Truss Design With Dynamic Grouping of Members." Journal of Mechanical Design **119**(3): 388-394.

Sigmund, O. and J. Petersson (1998). "Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima." Structural optimization **16**(1): 68-75.

Soh, C. and J. Yang (1996). "Fuzzy Controlled Genetic Algorithm Search for Shape Optimization." Journal of Computing in Civil Engineering **10**(2): 143-150.

Starr, M. (2015). "World's first 3D-printed apartment building constructed in China." Retrieved 09/02/2015, 2015, from http://www.cnet.com/news/worlds-first-3d-printed-apartment-building-constructed-in-china/.

Tashakori, A. and H. Adeli (2002). "Optimum design of cold-formed steel space structures using neural dynamics model." Journal of Constructional Steel Research **58**(12): 1545-1566.

Theano. (2015). "Theano 0.7 Documentation." Retrieved August 8, 2015, from http://deeplearning.net/software/theano/.

Topping, B. (1983). "Shape Optimization of Skeletal Structures: A Review." Journal of Structural Engineering **109**(8): 1933-1951.

Torabi, P., M. Petros and B. Khoshnevis (2014). "Selective Inhibition Sintering: The Process for Consumer Metal Additive Manufacturing." 3D Printing and Additive Manufacturing **1**(3): 152-155.

Vasanwala, S. A., J. A. Desai and H. S. Patil (2008). "BP And RBF Neural Networks For Predicting Minimum Weight of Double-Layer Space Grids." International Journal of Applied Engineering Research **3**(7).

Wang, G. G. and S. Shan (2006). "Review of Metamodeling Techniques in Support of Engineering Design Optimization." Journal of Mechanical Design **129**(4): 370-380.

Wang, L., W. Shen, H. Xie, J. Neelamkavil and A. Pardasani (2002). "Collaborative conceptual design—state of the art and future trends." Computer-Aided Design **34**(13): 981-996.

Weiss, S. M. and C. A. Kulikowski (1991). "Computer systems that learn: classification and prediction methods from statistics." Neural Networks, Machine Learning, and Expert Systems.

Wohlers, T. (2012). Additive manufacturing state of the industry annual worldwide progress report. Fort Collins, Colorado, Wohlers Associates.

Wright, G. R. (2009). Ancient Building Technology, Volume 3: Construction (2 vols), Brill.

Xiaohui, H. and R. Eberhart (2002). Multiobjective optimization using dynamic neighborhood particle swarm optimization. Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on.

Xie, Y. M. and G. P. Steven (1993). "A simple evolutionary procedure for structural optimization." Computers & Structures **49**(5): 885-896.

Zhenghao, Y. and K. Behrokh (2009). "Geometric conformity analysis for automated fabrication processes generating ruled surfaces: demonstration for contour crafting." Rapid Prototyping Journal **15**(5): 361-369.